Combinatorial Algorithms

Exercises

1 Generation of Basic Combinatorial Objects

1.1 Basic Examples

1. Let T be an arbitrary tree on n vertices.

(a) What is the size of a maximum cut in T? How many cuts attain this maximal size?

(b) How is the size of a random cut (i.e., a cut obtained by letting each vertex belong to the set with a probability of 1/2, independently of other vertices) distributed?

2. We choose a random graph G, using the $G(n,p)$ model. Namely, we start with a vertex set $\{v_1, v_2, \ldots, v_n\}$ of size n, and let each (v_i, v_j) with $i \neq j$ be an edge with probability p. Now we choose a random cut. Find the expected value of its size.

3. Consider the sum:

$$\sum_{(\varepsilon_1,\ldots,\varepsilon_n)\in\{0,1\}^n} f(\varepsilon_1,\ldots,\varepsilon_n).$$

(a) Show that it may be calculated in $O(n)$ time in each of the following cases:

(i) $f(\varepsilon_1,\ldots,\varepsilon_n) = c_1\varepsilon_1+\ldots+c_n\varepsilon_n$ for some constants $c_1,\ldots,c_n$.

(ii) $f(\varepsilon_1,\ldots,\varepsilon_n) = a^{c_1\varepsilon_1+\ldots+c_n\varepsilon_n}$ for some constants $a>0$ and $c_1,\ldots,c_n$.

(iii) $f(\varepsilon_1,\ldots,\varepsilon_n) = \cos(c_1\varepsilon_1+\ldots+c_n\varepsilon_n)$ for some constants $c_1,\ldots,c_n$. (Hint: $\cos\theta = (e^{i\theta}+e^{-i\theta})/2$.)

(b) How fast can you calculate the sum above if $f(\varepsilon_1, \dots, \varepsilon_n) = P(c_1\varepsilon_1 + \dots + c_n\varepsilon_n)$ for some polynomial P and constants $c_1, \dots, c_n$?

4. Put:

$$M_n = \frac{1}{2^n} \sum_{(\varepsilon_1,\dots,\varepsilon_n)\in\{0,1\}^n} \sqrt{\sum_{i=1}^{n} \varepsilon_i}\,.$$

(a) Express M_n is the form $M_n = E(h(X))$, where X is a random variable whose distribution belongs to some well-known family of distributions and $h : \mathbf{R} \longrightarrow \mathbf{R}$ is a suitable function.

(b) Use the inequality $E^2(Y) \le E(Y^2)$, which holds for any random variable Y, to deduce that $M_n \le \sqrt{n/2}\,$.

(c) Show that $M_n = \sqrt{\frac{n}{2}} \cdot (1 - o(1))$. (Hint: Use Chebyshev's inequality to show that for "most" n-tuples $(\varepsilon_1, \dots, \varepsilon_n)$ we have $\sum_{i=1}^{n} \varepsilon_i > \frac{n}{2} - n^{2/3}\,$.)

1.2 Generation of Subsets

5. We need to go over a certain family of subsets of $\{a_1, \dots, a_n\}$. The following algorithm has been suggested: Go over all subsets using the binary expansion approach, and for each of them test whether it belongs to the required family or not. Find the time complexity of the algorithm, and suggest improvements to the algorithm if possible, if the family consists of all

(a) subsets of even size;

(b) subsets of size $\lfloor n/2 \rfloor$;

(c) subsets not containing adjacent elements (i.e., if a_i belongs to the subset for some i, then a_{i+1} does not).

6. In our construction of a Gray code of order n, we may order the coordinates in any way, which gives in principle $n!$ Gray codes. Are these codes all distinct from each other?

7. Consider our construction of the Gray code $G(n)$. Design an efficient algorithm (polynomial in n) that, given an integer i in the range $[0, 2^n - 1]$, finds the ith vector G_i in the code.

8. Consider the Tower of Hanoi Problem. Let M_n be the sequence of length $2^n - 1$ recording which disk moves at each step of the process if there are n disks. Prove that $M_n = T_n$, where T_n is the sequence, introduced in class, of bits changing when going over all elements of $\{0,1\}^n$ by the Gray code.

9. Consider the set

$$B = \{0, 1, \ldots, d_1 - 1\} \times \{0, 1, \ldots, d_2 - 1\} \times \ldots \times \{0, 1, \ldots, d_k - 1\},$$

where $d_1, d_2, \ldots, d_k \geq 2$ are integers. Two elements $(x_1, \ldots, x_k)$ and $(y_1, \ldots, y_k)$ in B are *adjacent* if they are at a distance of 1 apart when we consider the elements in each coordinate j to ordered cyclically. Namely, $(x_1, \ldots, x_k)$ and $(y_1, \ldots, y_k)$ in B are adjacent if $y_l = x_l \pm 1 (\operatorname{mod} d_l)$ for some $1 \leq l \leq k$ and $y_i = x_i$ for every $i \neq l$. A *Gray code* for B is a sequence of elements of B, containing a unique occurrence of each element of B, in which consecutive entries are adjacent.

(a) Show that, for every k-tuple $(d_1, \ldots, d_k)$, the set B admits a Gray code.

(b) Characterize those k-tuples for which B admits a Gray code with the additional property that the last element of the sequence is adjacent to the first.

10. The solution presented in class to the problem of selecting a random subset of an n-element set (or, equivalently, a sequence of length n over $\{0,1\}$) involves n selections of random numbers. The following algorithm, which requires a single selection of a random number, has been suggested: Select a random number $r \in [0,1)$, multiply it by 2^n and take the integer part $s = \lfloor 2^n r \rfloor$. The bits of s form a random sequence as required.

(a) Is the algorithm theoretically correct?

(b) What do you expect the algorithm to yield in practice for large n (say, $n = 100$)?

1.3 Generation of Permutations

11. Let P be an arbitrary fixed subset of the set of all permutations of $\{1, 2, \ldots, n\}$. We want to design an algorithm which, given a permutation σ, returns the smallest permutation (according to the lexicographic order) in P which is greater or equal to σ (and returns the smallest permutation in P if σ is greater than all elements of P).

(a) The following algorithm has been suggested: Start with $\pi = \sigma$. While $\pi \notin P$, replace π by its successor. Analyze this algorithm in the average case and the worst case if P is the set

   (i) D_n of all derangements (permutations σ with $\sigma(i) \neq i$ for each i);

   (ii) ND_n of all non-derangements.

(b) Suggest worst-case polynomial time algorithms for the problem for both D_n and ND_n. Analyze their performance.

12. A permutation $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ is *cup-shaped* if $\sigma_1 > \sigma_2 > \ldots > \sigma_k < \sigma_{k+1} < \ldots < \sigma_n$ for some $1 \leq k \leq n$. Design an algorithm which goes over all cup-shaped permutations in linear time (in the number of such permutations).

13. In the algorithm for traversing the set of all permutations with minimal changes, presented in class, at each stage exactly two elements change their locations. How many elements on the average (asymptotically) change their locations at each stage in the algorithm which traverses the permutations in lexicographic order?

14. Which permutation is encountered last when we traverse all permutations with minimal changes, according to the algorithm presented in class?

1.4 The Coupon Collector's Problem

15. The three algorithms below have been suggested for selecting a random permutation of $1, 2, \ldots, n$. For each of them, determine whether it is correct (i.e., chooses each permutation with the same probability $1/n!$) and, if so, find the average number of selections of random integers required to obtain a random permutation. How does this average behave as $n \to \infty$?

(a) Choose n random integers between 1 and n until the chosen n-tuple forms a permutation.

(b) We need to select $\sigma_1, \sigma_2, \ldots, \sigma_n$. The numbers are selected one by one. At the k-th stage, $k = 1, 2, \ldots, n$, select a random integer between 1 and n repeatedly until it is distinct from all those selected before, and set σ_k as this integer.

(c) Start with the permutation $(1, 2, \ldots, n)$. Repeatedly select two random integers i and j between 1 and n, and swap σ_i and σ_j. Repeat this procedure l times, where l is sufficiently large.

16.

(a) How many children does a family have on the average until it has both a boy and a girl?

(b) Due to the low birthrate in Europe, the European Community is considering special benefits to families with at least k boys and l girls (where k and l are still to be determined). Denote by c_{kl} the number of children a family will have on the average to meet this criterion. Find a recurrence formula for c_{kl}.

17.

(a) Find the average number of coupons drawn in the Coupon Collector's Problem if $n = 2$ and the probabilities of getting the two coupon types are not equal, but are rather p and $q = 1-p$.

(b) Same for $n = 3$ and probabilities p_1, p_2, p_3 (with $p_1+p_2+p_3 = 1$).

1.5 Generation of Subsets of a Fixed Size

18. Suppose we want to go over the set of all subsets of size k of $\{1, 2, \dots, n\}$ (where $1 \le k \le n-1$) with minimal changes. However, we restrict the notion of a minimal change to refer only to interchanges of consecutive numbers, i.e., where some number s is removed from the subset and $s+1$ is added or vice versa.

(a) Show that, if n is odd and $\binom{n}{k}$ is even, then it is impossible to go over the set with minimal changes. (Hint: How many leaves can a graph with a Hamiltonian path have?)

(b) Show that there exist infinitely many pairs (n, k) for which it is possible to go over the set with minimal changes.

19. Consider the set of all subsets of size either k or $k+1$ of a set of size n. A *minimal change* of a subset consists of either removing an element from the subset or adjoining an element to it.

(a) Prove that, if $n \ge 3$ and $n \ne 2k+1$, it is impossible to go over the set with minimal changes.

(b) Show that, for $(n, k) = (1, 0), (2, 0), (3, 1), (5, 2)$, it is indeed possible to go over the set with minimal changes.

(c) Let $n = 2k+1 \ge 3$. Suppose we go over all subsets of the given set using the Gray code, shown in class. Now omit all subsets of sizes other than k and $k+1$. Show that the remaining sequence does not yield a traversal of our set with minimal changes.

(d) Prove or disprove: If $n = 2k+1$, then it is possible to go over the set with minimal changes. (This is the so-called Mid-Levels Conjecture; see, for example, `http://teaching.csse.uwa.edu.au/units/CITS7209/lecture01.pdf`.)

20. A *legal expression in parentheses* is a word in the language of balanced paretheses. (Namely, it is a word over the alphabet $\{(,)\}$, in which the total number of parentheses of the two types is equal and in every prefix of which the number of right parentheses does not exceed that of left parentheses.)

(a) For any positive integer n, denote by a_n the number of legal expressions in parentheses of length $2n$. Prove that

$$a_n = a_0a_{n-1} + a_1a_{n-2} + a_2a_{n-3} + \ldots + a_{n-1}a_0, \qquad n = 1, 2, \ldots .$$

(b) Employing generating functions, conclude from part (a) that $a_n = \binom{2n}{n}/(n+1)$ for each n.

(c) Note that a legal expression in parentheses of length $2n$ is uniquely determined by the set (of size n) of locations of the left parentheses. Thus, given any algorithm for traversing the set of all subsets of size k of a set of size n, we may use it (with $2n$ and n instead of n and k, respectively) to traverse the set of legal expressions in parentheses of length $2n$ (by going over all expressions with n left and n right parentheses and omitting the illegal ones). Suppose the given algorithm for traversing all subsets of size k is linear. Analyze the suggested algorithm for traversing all legal expressions in parentheses. (Assume that you can check the legality of an expression in time $O(1)$.)

(d) Develop a linear time algorithm for traversing all legal expressions in parentheses of length $2n$ in lexicographic order.

(e) Using parts (a) and (b), develop an algorithm of linear expected time for selecting a random legal expression in parentheses of length $2n$.

21. A subset of $\{1, 2, \ldots, n\}$ is *sparse* if it contains no two adjacent numbers.

(a) Suppose a linear time algorithm for traversing the set of all subsets of $\{1, 2, \ldots, n\}$ of size k (for every n and k) is given. Consider the following algorithm for traversing the set of all sparse subsets of $\{1, 2, \ldots, n\}$ of size k: Go over all subsets of size k and omit those which are not sparse. Is the suggested algorithm linear? If yes – prove it, if not – explain why not and suggest a linear time algorithm.

(b) Develop an algorithm for selecting a random sparse subset of size k which works in time $O(k)$.

1.6 Non-Uniform Random Selections

22. Let $S = \{1, 2, \ldots, n\}$, and let P be a probability measure on 2^S. It is required to select a random subset of S according to P. Suppose that an algorithm $\mathcal{A}$ for selecting a uniformly random subset of S is given. We employ the following algorithm for selecting a P-distributed subset: We select a uniformly random subset A of S using $\mathcal{A}$. Then we accept A with probability $P(A)/p_{\max}$ and reject it with probability $1 - P(A)/p_{\max}$, where

$$p_{\max} = \max_{A \subseteq S} P(A).$$

How many times (asymptotically) do we have to invoke $\mathcal{A}$ on the average to select one P-distributed subset for the following measures P?

(a) $P(A) = 0$ if $|A|$ is odd, and $P(A)$ is the same for all subsets A of even size.

(b) Subsets of odd size have twice the probability of subsets of even size.

(c) The probability of a subset of size k is proportional to 3^k.

(d) The probability of a subset of size k is proportional to $\binom{n}{k}$.

(e) The probability of a subset of size k is inversely proportional to $\binom{n}{k}$.

(f) The probability of a subset is proportional to the sum of its elements.

(g) The probability of a subset is proportional to the sum of the squares of its elements.

(h) The probability of a subset is proportional to its maximal element (where $P(\emptyset) = 0$).

(i) The probability of a subset is proportional to its minimal element (where $P(\emptyset) = 0$).

(j) The probability of a subset of size k is proportional to $k(k-1)$.

23. Design algorithms for selecting a P-distributed element in parts (d), (e), (f) and (j) of Question 22. In your algorithms, only one random set may be selected. (Hints: In (d), you will need to select a subset of size n of a set of size $2n$. In (j), count in two ways the number of possibilities for choosing a subcommittee of any size, and 2 distinguished members of this subcommittee, say a chairman and a secretary, out of a committee of size n.)

24. Similarly to Question 22, we want to select random permutations $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ of $\{1, 2, \ldots, n\}$ acccording to some probability measure P on the set of all permutations S_n. An algorithm $\mathcal{A}$ for selecting a uniformly random permutation σ is given, and we employ it analogously to Question 22. How many times (asymptotically) are we going to invoke $\mathcal{A}$ on the average to select a single P-distributed permutation for the probability measures P determined by the following conditions?

(a) $P(\sigma)$ is proportional to σ_1.

(b) $P(\sigma)$ is proportional to $\sigma_1 + \sigma_n$.

(c) $P(\sigma)$ is proportional to σ_1^2.

(d) $P(\sigma)$ is proportional to $\sigma_1 + 2\sigma_2 + \ldots + n\sigma_n$.

(e) $P(\sigma)$ is proportional to the number of indices $i, 1 \leq i \leq n$, for which $\sigma_i = i$.

(f) $P(\sigma)$ is proportional to the number of indices $i, 1 \leq i \leq n$, for which $\sigma_i \neq i$.

25. Design algorithms for selecting a P-distributed element in parts (a) and (d) of Question 24. In your algorithms, only one random set may be selected.

1.7 Generation of Partitions

26. Let $P(n, k)$ be the set of partitions of n whose maximal component is k.

(a) Modify the algorithm presented in class, for traversing the set of all partitions of n in lexicographic order, to a linear time algorithm for traversing $P(n, k)$ in lexicographic order.

(b) Consider the algorithm presented in class for traversing the set of all partitions of n in vocabulary order. Explain why it basically solves also the problem of traversing $P(n, k)$ in vocabulary order.

27. Denote by $p(n)$ the number of partitions of n (where we agree that $p(0) = 1$) and by $p(n,k)$ the number of those partitions of n whose maximal component is k.

(a) Denote by f the generating function of the double sequence $(p(n,k))_{n,k=0}^{\infty}$, namely

$$f(x,y) = \sum_{n=0}^{\infty}\sum_{k=0}^{\infty} p(n,k)x^n y^k.$$

Prove that f satisfies the functional equation

$$f(x,xy) = (1-xy)f(x,y).$$

(b) Let g be the generating function of the sequence $(p(n))_{n=0}^{\infty}$. Express g in terms of f.

(c) Show that $p(n+1) > p(n)$ for $n \geq 1$.

(d) Show that $p(n+1,k) \geq p(n,k)$ for each n and k.

(e) For each fixed k, find a polynomial Q such that $p(n,k) = \Theta(Q(n))$.

(f) Prove that $p(n,k) \leq \binom{n}{k}$ for each n and k.

(g) Use the preceding part to obtain an upper bound on $p(n)$.

(h) Obtain the upper bound on $p(n)$, obtained in the preceding part, directly (i.e., without using $p(n,k)$).

(i) Prove that $p(n) \geq 2^{C\sqrt{n}}$ for every $n \geq 2$ for an appropriate constant $C > 0$. (Hint: Restrict yourself to partitions using only some of the possible integers, and such that each component, except perhaps for 1, appears at most once.)

28. Consider the sum-of-divisors function σ.

(a) Prove σ is a multiplicative function (in the sense of Number Theory), namely that $\sigma(mn) = \sigma(m)\sigma(n)$ for relatively prime positive integers m, n.

(b) Deduce the following explicit formula for σ: If $n = p_1^{e_1}p_2^{e_2}\ldots p_k^{e_k}$ is the prime-power factorization of n, then

$$\sigma(n) = \frac{p_1^{e_1+1}}{p_1-1}\cdot\frac{p_2^{e_2+1}}{p_2-1}\cdot\ldots\cdot\frac{p_k^{e_k+1}}{p_k-1}.$$

1.8 Generation of Set Partitions

29. For integers $n \geq k \geq 1$, denote by $\mathcal{P}(n)$ the set of all partitions of the set $\{1, 2, \ldots, n\}$, and by $\mathcal{P}(n, k)$ its subset consisting of all partitions into k components. Put $a_n = |\mathcal{P}(n)|$ and $a_{n,k} = |\mathcal{P}(n, k)|$.

(a) Prove that for every fixed k we have $a_{n,k} = \Theta(k^n)$.

(b) Conclude from the previous part that the generating function of the sequence (a_n) converges only at the point 0.

(c) Prove that for every fixed k we have $a_{n,n-k} = \Theta(n^d)$ for an appropriate integer d.

(d) Find an explicit formula for $a_{n,2}$.

(e) Same for $a_{n,3}$.

(f) Same for $a_{n,n-1}$.

(g) Same for $a_{n,n-2}$.

(h) Prove that $n^{c_1 n} \leq a_n \leq n^{c_2 n}$ for suitable constants $c_2 > c_1 > 0$ for all sufficiently large n.

(i) Consider the order on $\mathcal{P}(n)$ according to which the algorithm given in class produces the partitions. Design an algorithm which, given a partition in $\mathcal{P}(n, k)$, produces the next partition belonging to $\mathcal{P}(n, k)$ (or reports that the given partition is the last in $\mathcal{P}(n, k)$). The algorithm should work in time $O(n)$, with the implicit constant independent of k.

1.9 Permanents

30. Prove that the permanent, considered as a function of a single row of the matrix (all other rows being held arbitrarily fixed), is a linear function.

31.

(a) Suppose that A consists of square blocks "along the diagonal", any entries to the right and above these blocks, and 0-s at the bottom left part. Express $\mathrm{Per}(A)$ in terms of the permanents of the blocks.

(b) Find a formula for the permanent of an upper triangular matrix.

32. What is the effect of exchanging the order of the rows of a matrix on its permanent?

33. Let A be an $n \times n$ matrix, in which each entry is selected uniformly randomly from the interval $[0, 1]$, independently of all other entries.

(a) Find the expected value of $\mathrm{Per}(A)$.

(b) Find the expected value of $\det(A)$.

34. Let $A = (x_{ij})_{i,j=1}^n$ be a generic $n \times n$ matrix (namely, a matrix whose entries are variables, not constants).

(a) Show that, for $n = 2$, by changing the sign of some appropriate entries of A, one can obtain a matrix A' such that $\det(A') = \mathrm{Per}(A)$.

(b) Show that the analogous statement for any $n \geq 3$ is false.

1.10 Young Tableaux

35. Explain intuitively which shapes admit many Young tableaux and which admit relatively few for a given number of cells n. Test your conjecture on numbers n of the form $n = m(m + 1)/2$, and verify it for $m = 2, 3$.

36. Suggest an algorithm for enumerating all Young tableaux with n cells.

37. The following algorithm has been suggested for selecting a uniformly random Young table with n cells: Choose a uniformly random partition π of n, and then choose a uniformly random Young table of shape π. Demonstrate that the algorithm is wrong.

38. The number a_n of all Young tableaux with n cells is known to satisfy the recurrence:

$$a_{n+1} = a_n + na_{n-1}.$$

Can you use this formula to design an algorithm for choosing a uniformly random Young table with n cells (similarly to the way we have used the hook-length formula to choose a random Young table of a given shape)?