Massachusetts Institute of Technology | Handout 27
6.042J/18.062J: Mathematics for Computer Science | April 8, 1997
Professor David R. Karger

Problem Set 9

Due: April 15, 1996.

Reading: Rosen 1.7–1.8, 5.1–5.3, 7.5.

Problem 1. Suppose we have a baseball league with 21 teams, and we are given a schedule in which each team plays 14 games.

(a) Assuming all teams are in different cities, prove that we can arrange for each team to play the same number of home and away games).

(b) Prove we can split the season into two halves (one before and one after the all-star break) in which each team plays seven games.

Problem 2. Suppose we have a graph with edges $e_1, \ldots, e_m$. Consider the following algorithm:

initially, unmark every edge
for $i = 1$ to m
  if the endpoints of e_i are not connected by a path of marked edges
    mark edge e_i

Prove that when the algorithm finishes, the set of marked edges forms a spanning forest of the input graph. (Hint: it may help to consider the graphs consisting of the edges $e_1, \ldots, e_k$ for various values of k.)

Problem 3. Consider the following state machine on a graph G. A state is specified by labeling each vertex of G as *uncolored*, *red*, or *blue*. Initially, all vertices are uncolored except for one vertex v which is red.

make-blue(w)
  Precondition:
    a neighbor of w is red
  Effect:
    w becomes blue

make-red(w)
  Precondition:
    a neighbor of w is blue
  Effect:
    w becomes red

(a) Prove that if G is bipartite and the above code reaches a state where all vertices are colored, then the coloring is a legal coloring (that is, all neighbors of a red vertex are blue, and vice versa). You'll want to use some facts from the last homework to do this.

(b) For what kinds of bipartite graphs is it possible to reach a state in which all vertices are colored (brief explanation without proof sufficient)?

Problem 4. Which of the following assertions is true? Prove your answer.

1. If f is $O(g)$ then $\ln f$ is $O(\ln g)$.
2. If $\ln f$ is $O(\ln g)$ then f is $O(g)$.
3. If f is $O(g)$ then $1/g$ is $O(1/f)$.

Problem 5. A college loan officer offers you the following deal. He will give you \$25,000 at the start of each year for four years starting today to pay for your college tuition. In return, you must pay \$15,000 at the start of each year for fifteen years starting in four years. Suppose the annual interest rate is 5%. Is this a fair deal?

Problem 6. The integral method

(a) Find tight (Θ) asymptotic bounds for the summation $\displaystyle\sum_{i=n}^{\infty} \frac{1}{i^2}$

(b) Give tight (Θ) bounds for $\displaystyle\sum_{i=1}^{n} \frac{1}{\sqrt{i}}$

Problem 7. Find a closed form expression $f(n)$ for $\displaystyle\sum_{i=1}^{n} i^2 x^i$.

Problem 8. By manipulating summations, without recourse to induction, prove that

$$\sum_{k=1}^{n} H_k = (n+1)H_n - n$$

Problem 9. Apparently, a standard model used by lobster fishermen is the following: the number of lobsters caught in a given year is the average of the number caught in the previous two years. Let's try out this model in the following hypothetical fishing community — in year 0, there were 100 lobsters caught, and in year 1, there were 300 lobsters caught.

(a) Let $f(i)$ denote the number of lobsters caught in year i. Write a recurrence relation for $f(i)$ and obtain a closed-form solution.

(b) Now let's consider the following variation on this model: the number of lobsters caught in a given year is the average of the number caught in the previous *three* years. To start things off, suppose that 100 were caught in year 0, 300 in year 1, and 200 in year 2.

Let $g(i)$ denote the number of lobsters caught in year i under this model. Again, write a recurrence relation for $g(i)$ and obtain a closed-form solution.

Problem 10. Consider the following recurrence:

$$T(n) = aT(n/2) + n^2 \log_2 n, \quad T(1) = 1.$$

(a) Using plug and chug, derive a solution to this recurrence in summation form.

(b) Prove that the recurrence satisfies this summation form

(c) By manipulating sums, derive a (simple) closed form for the summation in the case $a = 2$.

(d) Derive a (messy) closed form for the summation when $a \neq 2$.

Optional Problem 11. "Finite Calculus" works with sums the way standard calculus works with integrals. Just like standard calculus has an inverse to the integral, namely the derivative, finite calculus has an inverse to summation, namely the *difference* operator

$$\Delta_x f(x) = f(x+1) - f(x).$$

We can modify the summation notation to be more like integration, writing

$$\Sigma_a^b g(x)\delta x = \sum_{x=a}^{b-1} g(x).$$

(a) Prove that $\Delta_x \Sigma_a^x f(y)\delta y = f(x)$ (analogue to $D\int_a^x f(y)dy = f(x)$).

(b) Prove that $\Sigma_a^b(\Delta_x g(x))\delta x = f(b) - f(a)$ (analogue to $\int_a^b (Df(x))dx = f(b) - f(a)$)

(c) Define $x^{\underline{m}} = x(x-1)(x-2)\cdots(x-m)$. Prove that $\Delta_x x^{\underline{m}} = mx^{\underline{m-1}}$

(d) Prove that $\sum_{k=0}^{n} x^{\underline{m}} = \frac{n^{\underline{m+1}}}{m+1}$