

Sample Midterm Exam

Note: the following is much longer than the real exam will be. Also, doesn't include all possible topics (e.g. estimating)

Open Book and Notes

(45) 1. Suppose that you have a bipartite graph $G = (L, R, E)$ and a current matching M which is not maximum. Let n_L be the number of vertices in L , n_R the number in R , and m the number of edges in E . Let G_L be the *layered graph* in G with respect to M , where the layer 0 of G_L has the unmatched vertices in L , the last layer has only unmatched vertices in R , and in general layer i has those vertices which are distance i from an unmatched vertex in L on a shortest augmenting path, and only the last layer k has unmatched vertices in R (thus k is the length of a shortest A-path with respect to M).

For each of the following statements say whether it is true or false and briefly justify your answer:

- b) All augmenting paths can be found in G_L in $O(n_L + n_R)$ time.
- c) An adjacency list is a good way to represent the edges leaving a layer in G_L which contains nodes from L .
- d) An adjacency list is a good way to represent the edges leaving a layer in G_L which contains nodes from R .
- e) If G_L has at most 3 vertices in each layer, then we will find at most 3 A-paths in G_L .
- g) There are at most $2n_L$ layers in G_L .

(20) 2. After students graduate from Medical school they are assigned to hospitals. Each graduating student can choose those hospitals he/she would like to be assigned to, and each hospital can choose which graduating students they are willing to hire.

Your goal is to assign students to hospitals so that each student is assigned to a hospital which they want, and which wants to hire them (or if such an assignment is impossible, assign as many students as possible). You may assume you have all the preference data for both students and hospitals. Assume there are u students and h Hospitals.

- a) Describe how to find an assignment of students to hospitals when each hospital wants to hire ONE student.
- b) Describe how to find an assignment of students to hospitals when the i th hospital wants to hire up to w_i students.
- c) What is the running time of your solution technique in case a) and in case b)?

(Note, the following problem is NOT appropriate for this year's class since we did different programs, but it illustrates a kind of question I might ask).

(20) 3. In the select program for assignment 1 the main bottleneck of the code was the loop which partitioned the array into those less than t and those greater than t

```

    m = L;
    for (i=L+1; i <= u; i++) /* this loop rearranges the elements in
the range L+1 .. u so that all those < t are at the front */
        if (x[i] < t)
            swap(++m,i,x);

```

This approach works in-place (it rearranges the data within the existing array). An alternative is to use auxiliary array(s). This has the advantages of not changing the input array, and simplifies the partition problem in some ways:

If we use a new array y which is of size $u - L + 1$ we can use the following code to put the small elements at the front of y and the large ones at the end (with t in the middle):

```

small = 0; /* next element smaller than $t$ goes here */
big = u-L; /* next element larger than $t$ goes here */
for (i=L+1; i <= u; i++)
    if (x[i] < t)
        y[small++] = x[i];
    else y[big--] = x[i];
y[small] = t;

```

What are the potential advantages and disadvantages of this change in approach? Would you expect it to speed up or slow down the overall run time? (justify your answer).

(25) 1. Suppose you have a directed graph G . For two specified vertices a and b you want to find the smallest set of *vertices* whose removal disconnects all directed paths between a and b . Describe how to do this and give the running time of our solution when G has n vertices and m edges.

(45) 1. Consider the flow network G below (note the numbers on the arcs are the capacities).

a) Give a maximum flow for this network.

b) Give a minimum cut in this network.

c) What is the longest possible simple augmenting path in G (that is, for any flow or preflow f , what is the maximum length of an augmenting path in G_f which doesn't repeat a vertex)? Justify your answer.

d) Let p be the correct answer to part (c) above. What can you conclude about a node whose height reaches p when running the preflow-push algorithm on G ?

(20) 2. Suppose there are k students who are looking for internships and c companies which are looking for interns. For the i th student let L_i be the list of companies which are interested in student i and which student i wants to work for. Let p_i be the maximum number of students the i th company is willing to hire.

Your goal is to assign students to companies so that each student is assigned to a company which they want, and which wants to hire them (or if such an assignment is impossible, assign as many students as possible).

- a) Describe a flow network which finds an optimal assignment and draw a picture of your network.
- b) Now suppose that each company is willing to hire at most 2 students from the same school (all other constraints are the same). Describe a flow network to solve this setting and draw a picture of your network (for convenience of description you can assume that there are h schools and students 1-4 are from school one, 5-8 are from school two, ...).
- c) What is the running time of your solution technique in case a) and in case b)?

(20) 3. At a high level the select algorithm works as follows:

Repeat

 Pick a split element t ;

 Partition the array into Small, those $< t$, t and Big, those $> t$;

 Continue on either Small or Big

Until t is the answer

The profile shows that the second step is the main hot spot. However, it doesn't tell you how much time is spent for different iterations.

- a) How would you determine how much time is spent in different iterations of the Repeat loop?
- b) Which iterations would you expect to be the main bottleneck(s)?
- c) Given your answer to b), how might you improve the run time for those iterations which are most expensive (assume you are already using an efficient partition routine, so improving that is not an issue)?