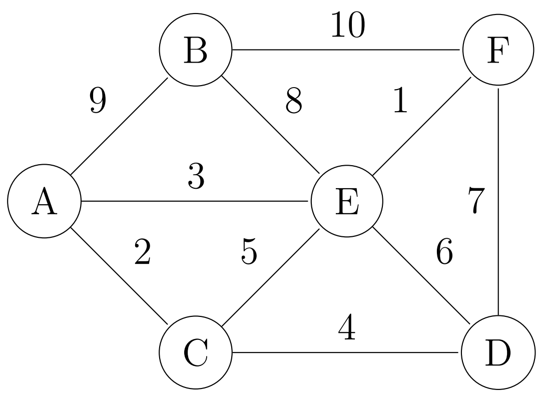
CMSC 351 Final Exam Fall 2020

Problem 1 (10 points). Illustrate the operations of radix sort on the following English words: TOP, POT, TOO, OPT, OPS, SOT, POP, SOP Just give the order of the words after each of the three passes.

Problem 2 (15 points). Consider the following undirected, weighted graph.



In the following problems you *must* use the edge weights to identify an edge.

(a) Assume that you use Kruskal's algorithm to find the Minimum Spanning Tree (MST). Show the order that the edges are included in the MST.

(b) Assume that you use Prim's algorithm to find the Minimum Spanning Tree, using vertex A as the source. Show the order that the edges are included in the MST.

(c) Assume that you use Dijkstra's algorithm to find the shortest paths from vertex A. Show the order that the edges are included in the shortest paths tree.

Problem 3 (15 points). Consider the area enclosed by the region above the x-axis, below the lines $y = x$ and $y = 2$, and left of the line $x = 4$. Assume that n points are uniformly distributed randomly inside it. (The n points can be represented by n pairs of real numbers $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.)

(a) Show that you can sort the points by their distance to the y-axis in average-case linear time. You can assume that bucket sort works in average-case linear time. You can assume that n is "nice" (such as n is a multiple of 6).

(b) Give the pseudo-code for your algorithm.

Problem 4 (15 points). Assume that you execute randomized selection. As a simplifying assumption, assume that at each iteration, not only is the pivot index, q, random, but the index of the value being searched for, k, is also random.

(a) Write a recurrence for the exact average number of comparisons.

(b) Use constructive induction to get an upper bound the average number of comparisons. Get the exact high order term.

Problem 5 (15 points). A 2-dimension mesh machine is a parallel computer with $P = k^2$ processors. Each processor is indexed (i, j) for $0 \leq i, j < \sqrt{P}$. Processor (i, j) (except those on an edge) has North, South, East, West neighbors $(i+1, j)$, $(i-1, j)$, $(i, j+1)$, $(i, j-1)$, respectively. In one parallel step, each processor can send exactly one value to a neighbor, and each processor can receive exactly one value from a neighbor. I.e., a pair of neighbors can communicate exactly one value and each processor can only be involved in one communication. Assume that we would like to find the sum of N values where each processor starts with N/P of the values.

(a) Give an efficient parallel algorithm to sum the values. You can describe the algorithm in clear English.

(b) What is the time as a function of N and P, in order notation?

(c) When is the algorithm efficient, in order notation? Show your work.

Problem 6 (30 points). At Kruskal Odd Offbeat Kollege (KOOK), all dorm rooms hold exactly three students. KOOK assigns roommates so that students will be as happy as possible.

We will use the matrix $H(i, j, k)$ for how happy students i, j, k are to room together. We want to maximize the sum

$$\sum_{\text{roommate triple } (i,j,k)} H(i, j, k)$$

The *optimization version* of the *roommate assignment problem* is given an $n \times n \times n$ matrix $H(i, j, k)$, where n is a multiple of 3, partition $\{1, \ldots, n\}$ into triples T to maximize the sum of $H(i, j, k)$ over all triples in T.

The *decision version* of the *roommate assignment problem* is given an $n \times n \times n$ matrix $H(i, j, k)$, where $3|n$, and a goal G, is there a partition of $\{1, \ldots, n\}$ into triples T so that the sum of $H(i, j, k)$ over all triples in T is at least G?

(a) Show that decision version of the Roommate Assignment Problem is in **NP**. Make sure to state what the certificate is, and to show that the verification is in polynomial time.

(b) Show that if you could solve the optimization version of the Roommate Assignment Problem is in polynomial time that you could also solve the decision version in polynomial time.

Write the pseudo-code using the following header:

```
function optimization(matrix H(i,j,k), int n)
```

which returns an array of n triples.

Make sure to analyze the run time. (You may not refer to Part (a) in this answer.)

(c) Show that if you could solve the decision version of the Roommate Assignment Problem in polynomial time that you could also solve the optimization version in polynomial time.

Write the pseudo-code using the following header:

```
function decision(matrix H(i,j,k), int n, int G)
```

which returns a Boolean.

Make sure to analyze the run time.