Print Name: ______________________ **ID:** ______________________

EC 504 – Fall 2021 – Practice Exam

This is sample of question. This probably la bit longer then the final. Equally valuable is to review the Homework Exercise for other possible exercises. Like the midterm it is close book and laptop but you may bring it 2 page cheat sheet to help recall details.

1. Answer True or False to each of the questions below. Each question is worth 2 points. Answer true only if it is true as stated, with no additional assumptions. Give an explanation to earn partial credit even if the answer is wrong.

   (a) In a forward star representation of a directed graph, the edges must be sorted by start node.

   (b) Suppose T1,T2 are minimum spanning trees for the same graph G. Then the largest edge weight of T1 must be the same weight as the largest edge weight of T2. .

   (c) In Bellman-Fords shortest path algorithm, the temporary distance labels D assigned to vertices which have not yet been scanned (i.e. left the work queue) can be interpreted as the minimum distance among all paths from the start vertex to the given vertices with intermediate vertices restricted to the vertices that have already been scanned (left the work queue).

   (d) Consider a minimum spanning tree T in a weighted, undirected graph G. Suppose we decrease the weight of a single edge that is in the minimum spanning tree T. Then, it is possible that T is no longer a minimum spanning tree in the modified graph.

   (e) Consider a minimum spanning tree problem in a connected, weighted graph with positive and negative weights which can include negative weight cycles. Then, both Prims and Kruskals algorithms can be used to find the minimum weight spanning tree in this graph.

   (f) For the master equation $T(n) = aT(n/b) + n^k$ with $\gamma = \log(a)/\log(b)$ the solution is always a sum of terms powers n^γ and n^k.

   (g) The adjacency matrix for a graphs $G(N, A)$ is always a symmetric matrix, i.e the same above and below the diagonal.

   (h) Given a sorted set of numbers the dictionary search will always perform have complexity $T(n) \in o(log(n))$.

   (i) If you have a minimum distance between (i,j) in an undirected graph $d(i, j) = d(i, k) + d(k, j)$ then either $d(i, k)$ or $d(k, j)$ is a minimum distance but not both.

   (j) The best union-find algorithm can perform a sequence random sequence of n unions and m finds in $Tn, m) \in O(n + m)$.

(k) There are exactly 5 distinct binary search trees that include the numbers 1,2 anad 3.

(l) The Johnson All to All distance algorithm for $G(N, A)$ introduces an augmented graph with an extra node and potential function to remove all the negative cycles so that Dijkstras one to all can be used.

(m) The scheduling algorithm with deadlines and unit tasks when you us the *Maximun Procrastination* method is an example of amortized analysis.

2. Consider a list of n elements with distinct values in the range $\{1, \ldots, n^2\}$. Identify which of the following sorting algorithms will produce a sorted list in worst case time $O(n \log n)$ (note I said $O()$, not $\Theta()$).

   (a) Counting (or Bin) sort
   (b) Radix sort with radix n
   (c) Merge sort
   (d) Insertion sort
   (e) Quicksort

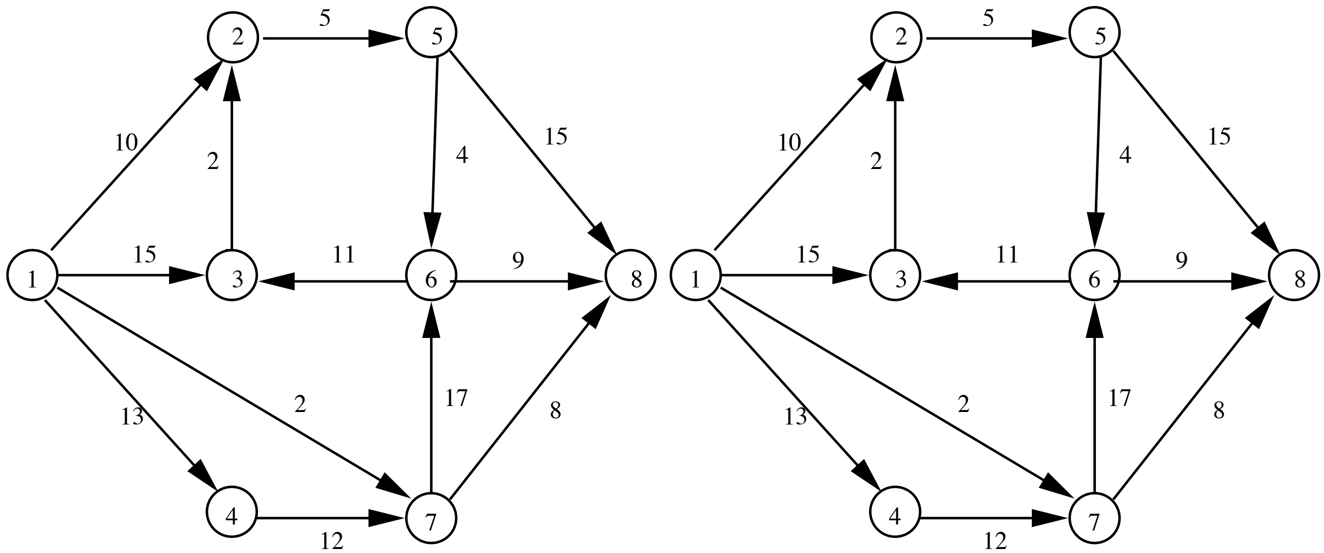

Figure 1:

3. Consider Fig. 1 as a directed, capacitated graph, where the numbers on an arc now indicate an arc's capacity to carry flow from node 1 to node 8. In the max-flow algorithm of Ford and Fulkerson, the key step is, once a path has been found, to augment the flow and construct the residual graph for the next iteration.

   (a) Suppose your program picks the first augmentation path to be $1 \to 7 \to 6 \to 3 \to 2 \to 5 \to 8$. Draw the augmented flow path on the left graph above and the residual graph above in the right side. What is the capacity of this path?

   (b) Now be smarter and beginning with a min hop $1 \to 2 \to 5 \to 8$ for the first path enumerate the additional paths that bring you to maximun flow. Draw **all** flow graphs and **all** the residual and after **each** augmentation. What is the maximum flow? Draw the minimum cut to show this right.

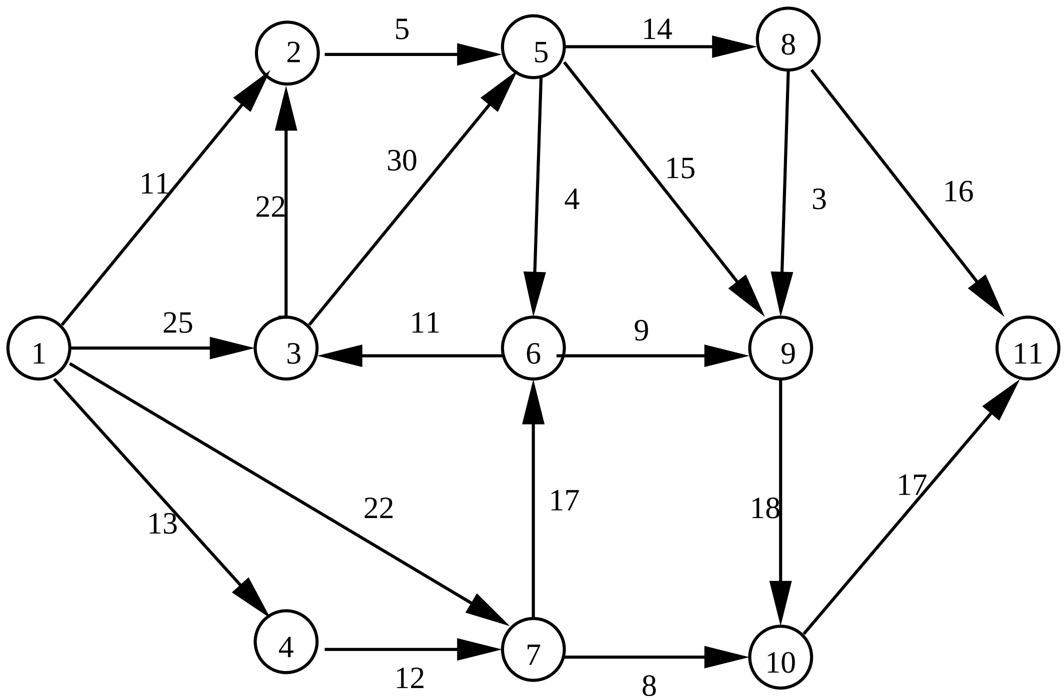

Figure 2:

4. Consider the weighted graph in Figure 2 but IGNORE THE ARROWS, TREATING IS AS AN UNDIRECTED GRAPH [1]

   (a) Use Dijkstra's algorithm to find the shortest paths from node 1 to all of the nodes. (Dijkstra's, like Prim's minimum spaning tree algorithm, adds one node at a time.)

   (i) Make a series of tables showing the values of the distance array $d(i)$ and the predecessor array $\pi(i)$ after each update

   (ii.) Draw the final tree on the figure with the final values of $d(i)$ and $p(i)$ at each node.

   (b) Repeat the exercise above except now use using Belllman-Ford this time (Bellman-Ford, like Kruskal's minimum spaning tree algorithm, adds one arc at a time.)

   (i.) Make a series of tables showing the values of the distance array $d(i)$ and the predecessor array $\pi(i)$ after each update.

   (ii.) Draw the final tree on the figure with the final values of $d(i)$ and $p(i)$ at each node.

   (c) Computed the minimum spanning tree considering all arcs to be bi-directional (in spite of the figure!) using Prim's algorithm starting form node 1, listing in order the total weight and predecessors as they are modified at each step.

   (d) Are the final trees in all these cases the same? Explain

[1]**Note in part a and b below you start with $d(1) = 0$, $d(i > 1) = \infty$ and $\pi(i) = -1$. The table at each step only needs to show the values of $d(i)$ and $\pi(i)$ that change!**

5. You are interested in compression. Given a file with characters, you want to find the binary code which satisfies the prefix property (no conflicts) and which minimizes the number of bits required. As an example, consider an alphabet with 8 symbols, with relative weights (frequency) of appearance in an average text file give below:

alphabet:	G	R	E	A	T	F	U	L
weights:	8	6	60	30	4	12	16	15

(a) Determine the Huffman code by constructing a tree with **minimum external path length**: $\sum_{i=1}^{8} w_i d_i$. (Arrange tree with smaller weights to the left.)

(b) Identity the code for each letter and list the number of bits for each letter and compute the average number of bits per symbole in this code. Is it less than 3? (You can leave the answer as a fraction since getting the decimal value is difficult without a calculator.)

(c) Give an example of weights for these 8 symbols that would saturate 3 bits per letter. What would the Huffman tree look like? Is a Huffman code tree always a full tree?

6. Consider searching in the "text" $T(1:N)$ of length $N = 25$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
a	b	c	a	a	b	c	a	b	a	b	c	a	b	a	b	c	a	b	c	a	b	a	b	c

for the following string of length $M = 8$ as an array $P(1:M)$ (i.e. $P(i), i = 1, 2, \cdots M$)

a	b	c	a	b	a	b	c

using the KMP algorithm.

(a) Give the prefix function for above string: That is the value of $\pi(i)$ for $i = 1, \cdots 8$.[2] It is useful to copy the pattern in $P(1:M)$ to slide it against $P(1:M)$ After a failure at $q+1$ you can safely shift by $\pi(q)$ before starting to search again. Note overlapping finds are ok!)

(b) Find all the instances of this pattern in the text. That is give the start index in the text for the aligned pattern. (You can do this even if you have not got the right prefex function. Slide $P(1:M)$ against $T(1:N)$.)

(c) Specify all the non-trivial shift (i.e grater than 1 unit) that occurred in the scan using the KMP algorithm.

[2] **Recall that the prefex function looks at the first q values of $P(1:q)$ and ask how far you can shift to match to have largest prefix match the suffix in these q terms.** $\pi(q) = MAX\{k < q$ **such that**$P(1:k) = P(1+q-k:q)\}$**.**

7. Consider the following knapsack problem. We have six objects with different values and Vi and sizes wi. The object values and sizes are listed in the table below:

Object number	1	2	3	4	5	6
Value	10	11	12	13	14	15
Size	1	1	2	3	4	3

(a) Suppose we are given a knapsack with total size 11, and you are allowed to use fractional assignments. What is the maximum value that fits in the knapsack for the above tasks?

(b) What is the maximum value of the tasks that fit entirely in the knapsack with size 11 — i.e. the integer knapsack.