Name: ______________________ UH ID: ______________

University of Houston

Final Exam Review

COSC 3320
Algorithms and Data Structures

Note

Read the Academic Honesty policy.

Name: ____________________ UH ID: ____________

Exercise 1: Old Stuff (5 Points)

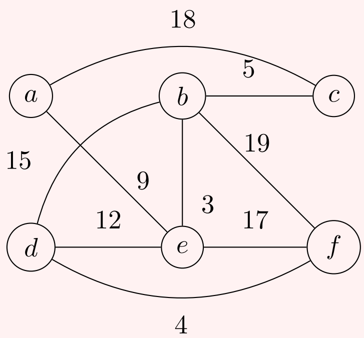
Review the following material from the first three exams:

1. exam 1
   - asymptotic notation (limit definition in particular)
   - decrease/divide and conquer
   - DC recurrence theorem
   - induction / recursion
   - sorting (know how the algos work, their runtime, etc)
   - median-of-medians (know how the algo works and runtime)
2. exam 2
   - maxsum, sweepy, hotel, matrix chain, LCS, LIS, etc.
   - topdown (memoized) vs. bottom-up
   - greedy algorithms and the exchange argument
3. exam 3
   - dfs, bfs, and their applications (e.g. cut vertices/edges, SCCs)

1 New Stuff (what will be emphasized on the final)

Exercise 2: Kruskal (5 Points)

Consider the graph G given below.



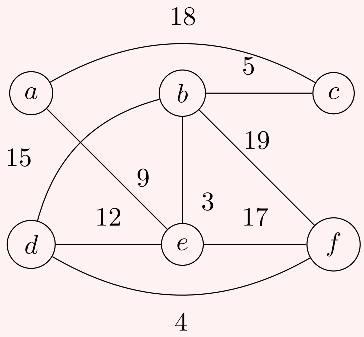
(a, c)	18
(a, e)	9
(b, c)	5
(b, d)	15
(b, e)	3
(b, f)	19
(d, e)	12
(d, f)	4
(e, f)	17

At each step of Kruskal's Algorithm

1. draw the minimum spanning forest
2. give a cut that justifies the inclusion of each edge

Exercise 3: Prim (5 Points)

Consider the graph G given below.



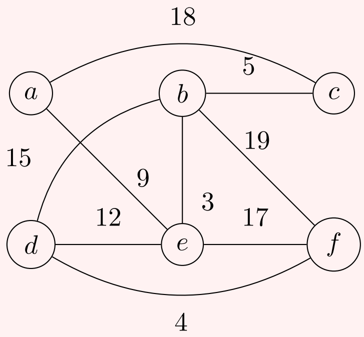
(a,c)	18
(a,e)	9
(b,c)	5
(b,d)	15
(b,e)	3
(b,f)	19
(d,e)	12
(d,f)	4
(e,f)	17

At each step of Prim's Algorithm

1. draw the minimum spanning tree
2. give *the* cut $(T, V-T)$ that justifies the inclusion of each edge

Exercise 4: SPT (10 Points)

Consider the graph G given below.



(a,c)	18
(a,e)	9
(b,c)	5
(b,d)	15
(b,e)	3
(b,f)	19
(d,e)	12
(d,f)	4
(e,f)	17

For each of the following, with vertex a as the root, draw the shortest path tree.

1. Dijkstra's Algorithm
2. the Bellman Ford Algorithm

Exercise 5: Cut Property (2 Points)

The Cut Property states that the minimum weight edge crossing a cut belongs to an MST.

1. Prove the cut property. You may assume the edge weights are unique.
2. Prove the correctness of Prim's algorithm, assuming the cut property. (Hint: prove by induction that every edge added to T belongs to the MST. After $n-1$ edges are added, what must T be?)

Exercise 6: Simple Path (2 Points)

A *simple* path is a path that does not revisit any vertices (equivalently, a path that contains no cycles).

Given a weighted digraph G as an adjacency matrix and distinct vertices u and v:

1. Give an $\mathcal{O}(n \cdot n!)$ algorithm to determine the shortest simple path from u to v
2. What restrictions can we place on G to improve this to polynomial time? Justify your answer.

Exercise 7: Shortest Path Table (1 Points)

Fill in the following table with the runtime of each algorithm used to solve each problem.

algorithm	runtime	
	SSSP	APSP
Dijkstra's		
Bellman Ford		
Floyd Warshall		