COS 226	Algorithms and Data Structures	Fall 2015

Final Exam Solutions

Grading note: The exam was graded out of 150 points. Grading on true/false and multiple-choice questions was as follows:

True / False:	$+1$ point if correct,	-1 point if incorrect,	0 points if left unanswered.
Multiple choice:	$+2$ points if correct,	-0.4 points if incorrect,	0 points if left unanswered.

1. Flow.

(a) 11 (2 points)

(b) SABEFT (3 points)

(c) 13 (2 points)

(d) ABEFT (3 points)

2. SPT.

(a) 1/2 point per correct answer

	`distTo[]`	`edgeTo[]`
0	0	–
1	24	0
2	5	0
3	6	2
4	34	5
5	14	3

(b) 4 (2 points)

(c) 4 (2 points)

(d) S (2 points)

3. TST.

(a) 1, 2, 5, 7 (1 point for correctly circling or not circling each one)

(b) A/18 is the left child of C, A/0 is the left child of T, C/29 is the middle child of A (2 points each)

4. KMP DFA.

(a) 1/2 point per correct answer

j	0	1	2	3	4	5
`pat.charAt(j)`	A	B	B	A	B	A
A	1	1	1	4	1	6
B	0	2	3	0	5	3

(b) ABABABB (2 points)

(c) 1, 5, 5, 4 (1 point each)

5. DAG.

(a) 13, 23, 24, 7, 20, 3 (3 points, partial credit if the vertices from the source iterator were not included.)

(b) 10 (3 points)

(c) $(V+E)^2$ (2 points)

(d) True (1 point)

(e) False (1 point)

6. Regex.

(a) ADAAC, BDC, BDAC (3 points total)

(b) 1, 2, 3, 5, 7, 10, 11, 12, 14 (2 points total)

(c) False, True, False, False, True, False (1 point each)

7. Huffman.

(a) FEEDDEAD (2 points)

(b) 3/8 (2 points)

(c) 1/8 (2 points)

(d) Only D (2 points)

(e) 3/4 (2 points)

8. Graph T/F.

False, True, False, False, True, True, False, True, False, False (1 point each)

9. Sort.

3, 1, 2, 1, 2, 2, 3 (2 points each)

10. G^2.

```
create a new graph G' with the same number of vertices as G

for each vertex u in G
    for each vertex w in adj(u) in G
        for each vertex v in adj(w) in G
            add edge (u,v) to G'
```

The two outer `for` loops together iterate over every edge, hence take time proportional to E. Copying an adjacency list takes time proportional to the length of that list, which is V in the worst case. Hence, this runs in $O(VE)$ time.

11. ST Analysis.

Remember that these asked for *worst case*. (2 points each)

(a) L lg N

(b) NL

(c) NL

(d) L

(e) RL

12. Reduction.

True, False, False, True, False, True, True, False, False, False (1 point each)

13. MST.

(a) Add the updated edge to the MST, which will create a cycle. Run a cycle detector on the augmented MST, to find the edges in that cycle. Loop over those edges, keeping track of the one having maximum weight. Remove that maximum-weight edge, restoring an MST.

Finding a cycle (using something like DFS) is normally $O(E)$, but the augmented MST has only V edges, hence runs in $O(V)$ time. Similarly, finding the max-weight edge in that cycle can be done in $O(V)$ time. So, the running time is $O(V)$.

(6 points)

(b) Do nothing, since this can't affect the MST. Constant time. (2 points)

(c) Do nothing, since this can't affect the MST. Constant time. (2 points)

(d) Remove the edge in question from the MST, which will disconnect it into two connected components. Run DFS from one of the endpoints of that edge, marking all vertices in one of those components. (The unmarked vertices form the other connected component.) Now loop over all edges, keeping track of the minimum-weight edge having one endpoint in the marked set and its other endpoint in the unmarked set. Add that minimum-weight edge, restoring an MST.

DFS to find a connected component will run in $O(E)$ time, as will finding the minimum-weight edge spanning the two connected components. So, the running time is $O(E)$.

(6 points)