COS 226	Algorithms and Data Structures	Fall 2011

Final Solutions

1. **Analysis of algorithms.**

   (a) $T(N) = \frac{1}{10} N^{5/3}$.

   When N increases by a factor of 8, the memory usage increases by a factor of 32. Thus, $T(N) = aN^b$, where $b = \log_8 32 = \lg 32 / \lg 8 = 5/3$. Since $T(1000) = 10000$, we have $10000 = a \times 1000^{5/3}$, which implies $a = \frac{1}{10}$.
   Remark: this is essentially the same question from the midterm.

   (b) B D F C B E A

   i. Single loop.
   ii. Similar to insertion sort.
   iii. Similar to enumerating all permutations.
   iv. Similar to mergesort since $f_1(N)$ takes linear time.
   v. $N + N/2 + N/4 + \ldots$
   vi. Similar to enumerating all subsets.
   vii. Similar to binary search.

2. **Graph search.**

   (a) `2 8 0 7 1 6 3 5 4`
   (b) `2 0 1 8 6 7 4 3 5`

3. **Minimum spanning trees.**

   (a) `1 2 3 4 5 8 11 15`
   (b) `1 5 2 4 8 3 11 15`
   The starting vertex must be either A or F (but it doesn't matter which).

4. **Shortest paths.**

(a) `3 0 10 5 2`

It is not necessary to run the algorithm from the beginning to deduce this because Dijkstra's algorithm considers the vertices in increasing order of distance from the source.

(b) The next vertex to relax is 4.

v	distTo[]	edgeTo[]
0	1.0	$3 \to 0$
1	17.0	$5 \to 1$
2	6.0	$5 \to 2$
3	0.0	*null*
4	7.0	$0 \to 4$
5	5.0	$10 \to 5$
6	11.0	$4 \to 6$
7	10.0	$4 \to 7$
8	8.0	$4 \to 8$
9	22.0	$4 \to 9$
10	3.0	$3 \to 10$

5. **String sorting.**

c	count[]	count[]	count[]
⋮	⋮	⋮	⋮
'a'	0	0	0
'b'	0	0	3
'c'	3	3	5
'd'	2	5	11
'e'	6	11	17
'f'	6	17	20
'g'	3	20	20
⋮	⋮	⋮	⋮

i	a[i]
0	blurb
1	climb
2	crumb
3	basic
4	cubic
5	freed
⋮	⋮
18	dwarf
19	cliff

6. **Substring search.**

	0	1	2	3	4	5	6	7	8	9	10
A	0	0	3	0	0	3	7	0	0	10	11
B	1	2	2	4	5	6	2	8	9	6	4
s	B	B	A	B	B	B	A	B	B	A	A

The answer is unique.

7. **Regular expressions.**

The NFAs for (i), (ii), (iii), and (iv) would not be constructed by our RE-to-NFA algorithm.

(i) There are two missing ϵ-transitions: $2 \to 6$ and $6 \to 2$.

(ii) The edges $2 \to 3$ and $7 \to 8$ should be match transitions; the edge $3 \to 4$ should be an ϵ-transition.

(iii) There are two missing match transitions: $2 \to 3$ and $4 \to 5$. Also, the edges $7 \to 8$ and $8 \to 9$ should be match transitions instead of ϵ-transitions.

(iv) The ϵ-transition $2 \to 6$ should instead be $0 \to 5$.

(v) Correct.

8. **Ternary search tries.**

`BD    C    CD    E    FD    JPG    PEGS`

9. **String symbol table implementation.**

BCDE	Find the value associated with a given string key in the data structure.	A. Unordered array.
CDE	Associate a value with a string key.	B. Ordered array.
CDE	Delete a string key (and its associated value) from the data structure.	C. Red-black BST.
BCE	Find the smallest string key in the data structure.	D. Separate-chaining hash table.
BCE	Find the smallest string key in the data structure that is greater than or equal to a given string.	E. Ternary search trie.
E	Find the string key in the data structure that is the longest prefix of a given string.	
E	How many string keys in the data structure starts with a given prefix?	

Can also be done with an ordered array (B) or a red-black BST (C) by calling `rank()` *twice, once with the prefix and once with the last character in the prefix incremented by one.*

10. **Data compression.**

(a) C A A C B

i. The frequency of A can be less than the frequency of B or it can be equal to the frequency of B.

ii. Since A and B are merged first, they have are symbols that have the smallest frequencies.

iii. Clearly $freq(D) \geq freq(A)$. Suppose $freq(D) = freq(A)$. Then, since $freq(D) \geq freq(C) \geq freq(A)$, we must have $freq(D) = freq(C) = freq(A)$. In this case, C and D would be merged (instead of C and $\{A, B\}$).

iv. If A, B, and C have frequency 1, then D could have frequency 2 or 3 and produce the same subtree.

v. If the frequency of E is strictly less than that of A, B, and C combined, then so is the frequency of D. Hence, D and E would be merged.

Note that if a character appears 0 times, then it will not appear in the Huffman trie.

(b)

LZW-encoded message	*original message or brief explanation why invalid*
`42 41 4E 82 41 80`	`B A N A N A`
`42 41 83 80`	not possible since `83` would not yet be in table
`41 42 81 82 80`	`A B A B B A`
`41 42 81 83 80`	`A B A B A B A` (see p. 843 of textbook)
`42 41 4E 44 41 4E 41 80`	not possible because of second occurrence of `41 4E`

11. **Maximum flow.**

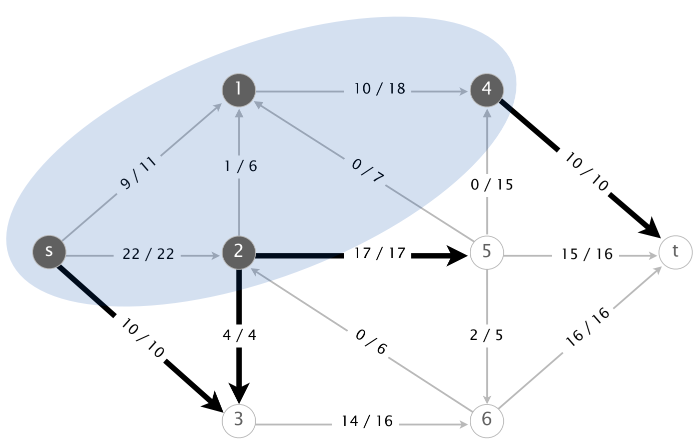
(a) $16 + 13 + 10 = 39$.

(b) $s \to 1 \to 2 \to 6 \to 5 \to t$.

(c) After using the previous augmenting path, there are no more augmenting paths. The value of the maximum flow is 41 (10 + 15 + 16).

(d) $\{s, 1, 2, 4\}$.

(e) $10 + 4 + 17 + 10 = 41$. *This equals the value of the maximum flow, as it should.*



12. **Algorithm design.**

(a) Build an edge-weighted graph G' containing all of the edges whose weight is strictly less than that of $e = v - w$. Find the connected components in G'. By the cut property, e is in the MST if and only if v and w are in different components.

(b) $E + V$.

13. **Reductions.**

(a) For each `i`, set `b[i]` equal to `a[i]` and `c[i]` equal to `-a[i]`.

`b[]`	-66	-30	70	99	-33	66	20	50

`c[]`	66	30	-70	-99	33	-66	-20	-50

(b) *Solution 1:* For each `j`, include the integer `b[j] + M` (guaranteed to be positive); for each `k`, include the integer `-(c[k] + 2*M)` (guaranteed to be negative). Any triple in `a[]` that sums to zero must include two positive integers and one negative integer.

`a[]`	1299	1700	1010	1014	997	999	1020
	-2999	-2019	-1996	-2600	-2030	-2020	

Solution 2: For each `j`, include the integer `10*b[j] + 1`; for each `k`, include the integer `-(10*c[k] + 2)`. To see why this works, consider the sum of the three integers modulo 10. Note: can replace 10 with any integer greater than or equal to 4.