solutions 3

1. (b) `A BE CFI DGJ HK L`

```
        A
   B          E
 C   F        I
D G  J
H K
L
```

(c) `A E B I C G D F H L K J`

```
   A       F        H
   E                L
B     I             K
C                   J
G
D
```

(d)

```
    A             B
    C          D  E  F
 G     H          I
 I   J   K
L M     N O
```

2. (a) Sublists of size 1 are already sorted, so no KC.

   $n-1$ KC to merge two lists with a two of n keys. The list is split evenly in two, so the two sublist sizes are floor and ceiling of $n/2$. Let $K(n)$ be the number of key comparisons performed by a mergesort of n keys.

(b) There are $n!$ different outputs, because there are $n!$ permutations of n keys. So there needs to be this many leaves in the algorithm's decision tree.

The number of KC required for any outcome will be the depth of the corresponding leaf in the decision tree. So we want to know the maximum depth of a node in a binary tree with t leaves, where t is at least $n!$. This is $\lg t$. But it is also an integer, so we can round up, so use the ceiling.

(c)

```
def k(n):
  L = [0,0]
  for j in range(2,n+1):
    L.append(j-1+ L[j/2] + L[(j+1)/2])
  return L[n]

def b(n):
  sum = 1.0
  for j in range(3,n+1):
    sum += math.log(j,2)
  return sum

for j in range(11):
  print j, k(j)
n = 10
for _ in range(6):
  n *= 10
```

```
print n, k(n)
```

```
2 1 1.0
3 3 2.58496250072
4 5 4.58496250072
5 8 6.90689059561
6 11 9.49185309633
7 14 12.2992080184
8 17 15.2992080184
9 21 18.4691330198
10 25 21.7910611147
100 573 524.76499329
1000 8977 8529.3980042
10000 123617 118458.143003
100000 1568929 1516704.17392
1000000 18951425 18488884.82
10000000 223222785 218108029.186
```

The maximum number returned by $k(n)$ is in $\Theta(n \log n)$, so has $\Theta(\log(n \log n) = \Theta(\log n + \log^2 n) = \Theta(\log n)$ bits. So the runtime is in $O(n \log n)$. Similarly, the numbers $k(n/2)$ to $k(n)$ each have $\Theta(\log n)$ bits, so the runtime is in $\Theta(n \log n)$.

For my data, k ranged from .86 to .96, and b from .79 to .94, and was growing, so .96 and .94 are probably lower bounds on k and b.

3. (a) H L K J F I E B A G D C
   D G C

```
C   A   F   J
G   E       K
D  B I      H
            L
```

(b) Yes, in D, there is a path from x to y, all of whose nodes are in S, by definition of scc.

Yes, same as above.

Yes. By 1st question above, there is a path from x to y in D, so there is a path from y to x in T, all of whose nodes are in S.

Yes. By 2nd question above, ...

Yes. For any x and y, we have shown there is a path from x to y in T, and from y to x, and in each case all nodes are in S. So S is strongly connected in T. Now, by repeating this argument, no proper subset of S can be strongly connected in T (because then it would be strongly connected in D, and so S would not be an scc of D), and similarly no proper superset of S can be strongly connected in T. So S is an scc of T.

Conclusion: sccs of a digraph are same as sccs of its transpose.

4. (a)

```
                      (0 9)
      (0 3)                           (5 9)
(0 0)       (2 3)          (5 7)        (9 9)
         (2 2) (4 3)       (5 6)   (8 7)
                        (5 5) (7 6)
0 9 [44, 88, 11, 0, 33, 99, 22, 77, 66, 55]
0 3 [11, 0, 33, 22, 44, 88, 99, 77, 66, 55]
0 0
2 3 [0, 11, 33, 22, 44, 88, 99, 77, 66, 55]
2 2
4 3
5 9 [0, 11, 22, 33, 44, 88, 99, 77, 66, 55]
5 7 [0, 11, 22, 33, 44, 77, 66, 55, 88, 99]
5 6 [0, 11, 22, 33, 44, 66, 55, 77, 88, 99]
5 5
7 6
8 7
9 9
```

(b) Partition does not preserve sublist order because to do usually requires lots of copying.

(c)

```
2 1 1.0 2.5
3 3 2.58496250072 5.33333333333
4 5 4.58496250072 8.66666666667
5 8 6.90689059561 12.4
6 11 9.49185309633 16.4666666667
7 14 12.2992080184 20.819047619
8 17 15.2992080184 25.4214285714
9 21 18.4691330198 30.246031746
10 25 21.7910611147 35.2706349206
100 573 524.76499329 763.683591897
1000 8977 8529.3980042 12151.7459962
10000 123617 118458.143003 167437.529266
100000 1568929 1516704.17392 2134719.2396
1000000 18951425 18488884.82 25952148.0645
10000000 223222785 218108029.186 305572926.541
```