University of Nevada, Las Vegas Computer Science 477/677 Fall 2025

Answwers to Examination November 25, 2025

1. True/False/Open (5 points each)

   (a) **F** Open hashing uses open addressing.

   (b) **F** Every mathematically defined problem can be solved by computation, although it might not have been done yet.

   (c) **T** Addition of two n-digit binary numerals can be done in $O(\log n)$ time with $O(n)$ parallel processors.

   (d) **T** Any connected graph with n vertices must have at least $n-1$ edges. (Hint: Draw figures.)

2. [20 points] Fill in the blanks.

   (a) A planar graph of 5 vertices can have no more than **9** edges.

   (b) A **perfect** hash function has no collisions.

   (c) The asymptotic expected height of a treap of n nodes is $\Theta(\log n)$.

   (d) The worst case asymptotic time complexity of Graham Scan is $\Theta(n \log n)$.

3. Write the asymptotic time complexity, in terms of n, of each of these code fragments.

   (i) [5 points]

   ```
   for(int i = 1; i < n; i = 2*i)
   ```

   Substitute $j = \log i$, $m = \log n$. You get

   ```
   for(int j = 0; j < m; j++)
   ```

   Answer is $\Theta(m) = \Theta(\log n)$

   (ii) [5 points]

   ```
   for(int i = 2; i < n; i = i*i)
   ```

   Just as in the problem above, substitute $j = \log i$ and $m \log n$: You get

   ```
   for(int j = 1; j < m; j=2*j)
   ```

   Now substitute $k = \log j$, $p = \log m$. You get

   ```
   for(int k = 0; k < p; k++)
   ```

   Answer is $\Theta(p) = \Theta(\log m) = \Theta(\log \log n)$

   (iii) [5 points]

   ```
   for(int i = 0; i*i < n; i++)
   ```

   Let $n = m^2 = m * m$. Then `i*i < n` is equivalent to `i < m`. We have:

   ```
   for(int i = 0; i < m; i++)
   ```

   Answer is $\Theta(m) = \Theta(\sqrt{n})$.

(iv) [5 points]

```
for(int i = 1; i < n; i++)
 for(int j = 1; j < n; j=j+j)
```

Let $k = \log j$. $\log(j + j) = \log(2j) = 1 + k$ We have:

```
for(int i = 1; i < n; i++)
 for(int k = 0; k < \log n; k++)
```

$\log n = \Theta(\ln n)$. Letting the continuous variables x and y represent the discrete variables i and k, we have: $\int_1^n \int_0^{\ln n} dydx = \int_1^n (\ln n)dx = (x \ln n)\Big|_1^n = \Theta(n \log n)$

(v) [5 points]

```
for(int i = 1; i < n; i++)
 for(int j = 1; j < i; j=j+j)
```

Using the substitution $k = \log i$ again we have:

```
for(int i = 1; i < n; i++)
 for(int k = 1; k < log i; k++)
```

We have: $\int_1^n \int_0^{\ln x} dydx = \int_1^n (\ln x)dx = (x \ln x - x)\Big|_1^n = n \ln n - n + 1 = \Theta(n \log n)$

(vi) [5 points]

```
for(int i = 1; i < n; i++)
 for(int j = i; j < n; j=j+j)
```

Using the substitution $k = \log i$ again we have:

```
for(int i = 1; i < n; i++)
 for(int k = \log i; k < \log n; k++)
```

$\int_1^n \int_{\ln x}^{\ln n} dydx = \int_1^n (\ln n - \ln x)dx = (x \ln n - x \ln x + x)\Big|_1^n = n \ln n - n \ln n + n - \ln n + 1 = \Theta(n)$

(vii) [5 points]

```
for(int i = 1; i < n; i = 2*i)
 for(int j = 1; j < i; j++);
```

Substitute $k = \log i$, $m = \log n$. Thus $n = 2^m$ and $i = 2^k$.

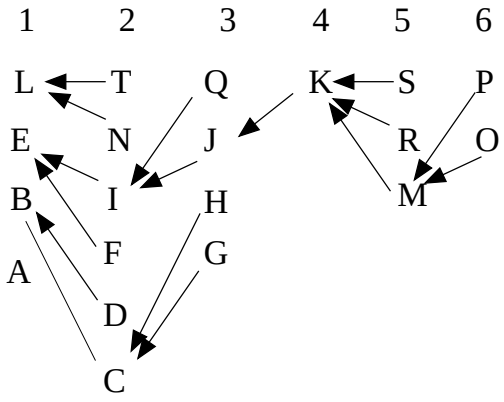
```
for(int k = 0; k < m; k++)
 for(int j = 1; j < i; p++);
```

Let x and y represent k and j. Recall that $\int 2^x dx = \frac{2^x}{\ln 2} + C$.

We have: $\int_0^m \int_1^{2^x} dydx = \int_0^m (2^x - 1)dx = \left(\frac{2^x}{\ln 2} - x\right)\Big|_{x=1}^m = \frac{2^m - 2}{\ln 2} - m + 1 = \Theta(2^m) = \Theta(n)$.

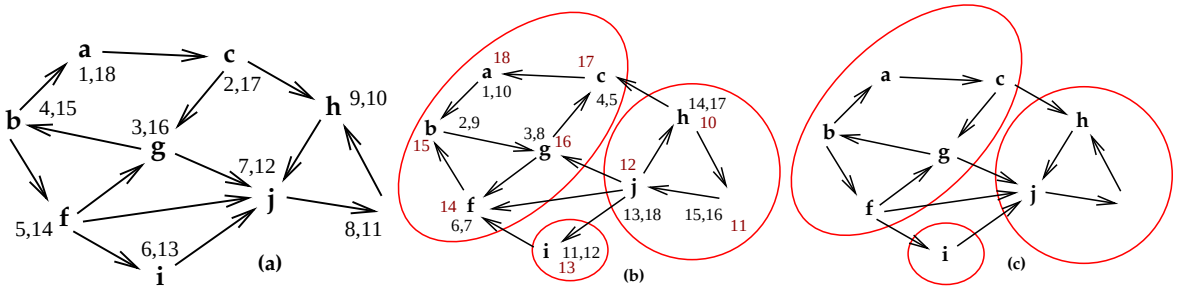
4. [20 points] Find a maximal increasing subsequence of the sequence LTNEIQJFKSBDRMPCOAHG. Use the algorithm given in class. Show your work.

L T N E I Q J F K S B D R M P C O A H G



There are many maximal increasing subsequences. The algorithm I gave in class returns EIJKMO.

5. [20 points] Find the strong components of the digraph illustrated in (a) below. Show the answer in (c) by circling the strong components.



6. Solve the following recurrences. Express the answers using Θ notation.

(i) [5 points] $F(n) = 2F(n/2) + n$

Master Theorem.
$A = 2$, $B = 2$, $C = 1$. The middle case, since $\log_B A = 1 = C$.
The answer is $F(n) = \Theta(n \log n)$

(ii) [5 points] $F(n) = F(n/3) + 1$

Master Theorem.
$A = 1$, $B = 3$, $C = 0$. The middle case, since $\log_B A = 0 = C$.
The answer is $F(n) = \Theta(\log n)$

(iii) [5 points] $F(n) = F(n/2) + F(n/3) + F(n/6) + 1$

Akra Brazzi method. $\alpha_1 = \alpha_2 = \alpha_3 = 1$, $\beta_1 = 1/2$, $\beta_2 = 1/3$, $\beta_3 = 1/6$, $\gamma = 0$
$\sum_{i=1} \alpha_i \beta_i^\gamma = 1 + 1 + 1 = 3 > 1$. Thus we need to find $\delta < \gamma$ such that
$\sum_{i=1} \alpha_i \beta_i^\delta = 1$ The correct choice is $\delta = 1$.
The answer is $F(n) = \Theta(n^\delta) = \Theta(n)$.

(iv) [5 points] $F(n) = F(n - \sqrt{n}) + 1$

We can apply the anti-derivative method since $\sqrt{n} = o(n)$. (That's "little oh," meaning that the increase of $\sqrt{n}$ is asymptotically strictly less than the increase of n, in other words, $\lim_{n\to\infty} \frac{\sqrt{n}}{n} = 0$.)
$F(n) - F(n - \sqrt{n}) = 1$
$\frac{F(n) - F(n - \sqrt{n})}{\sqrt{n}} = \frac{1}{\sqrt{n}}$
$F'(n) = n^{-\frac{1}{2}}$
$F(n) = \Theta(n^{\frac{1}{2}}) = \Theta(\sqrt{n})$

(v) [5 points] $F(n) = F(3n/5) + F(4n/5) + n$

Akra Brazzi method. $\alpha_1 = \alpha_2 = 1$, $\beta_1 = 3/5$, $\beta_2 = 4/5$. $\gamma = 1$
$\sum_{i=1} \alpha_i \beta_i^\gamma = 3/5 + 4/5 = 7/5 > 1$. We need to find $\delta < \gamma$ such that
$\sum_{i=1} \alpha_i \beta_i^\delta = 1$ The correct choice is $\delta = 2$, since $(3/5)^2 + (4/5)^2 = (9 + 16)/25 = 1$.
The answer is $F(n) = \Theta(n^\delta) = \Theta(n^2)$.

(vi) [5 points] $F(n) = 2F(4n/9) + F(7n/9) + n^2$

Akra Brazzi method. $\alpha_1 = 2$, $\alpha_2 = 1$, $\beta_1 = 4/9$, $\beta_2 = 7/9$. $\gamma = 2$
$\sum_{i=1} \alpha_i \beta_i^\gamma = 2(4/9)^2 + (7/9)^2 = 2(16/81) + 49/81 = (32 + 49)/81 = 1$.
This is the middle case, so we have
$F(n) = \Theta(n^\gamma) = \Theta(n^2)$.

(vii) [5 points] $F(n) = 3F(2n/3) + 3F(n/3) + n^3$

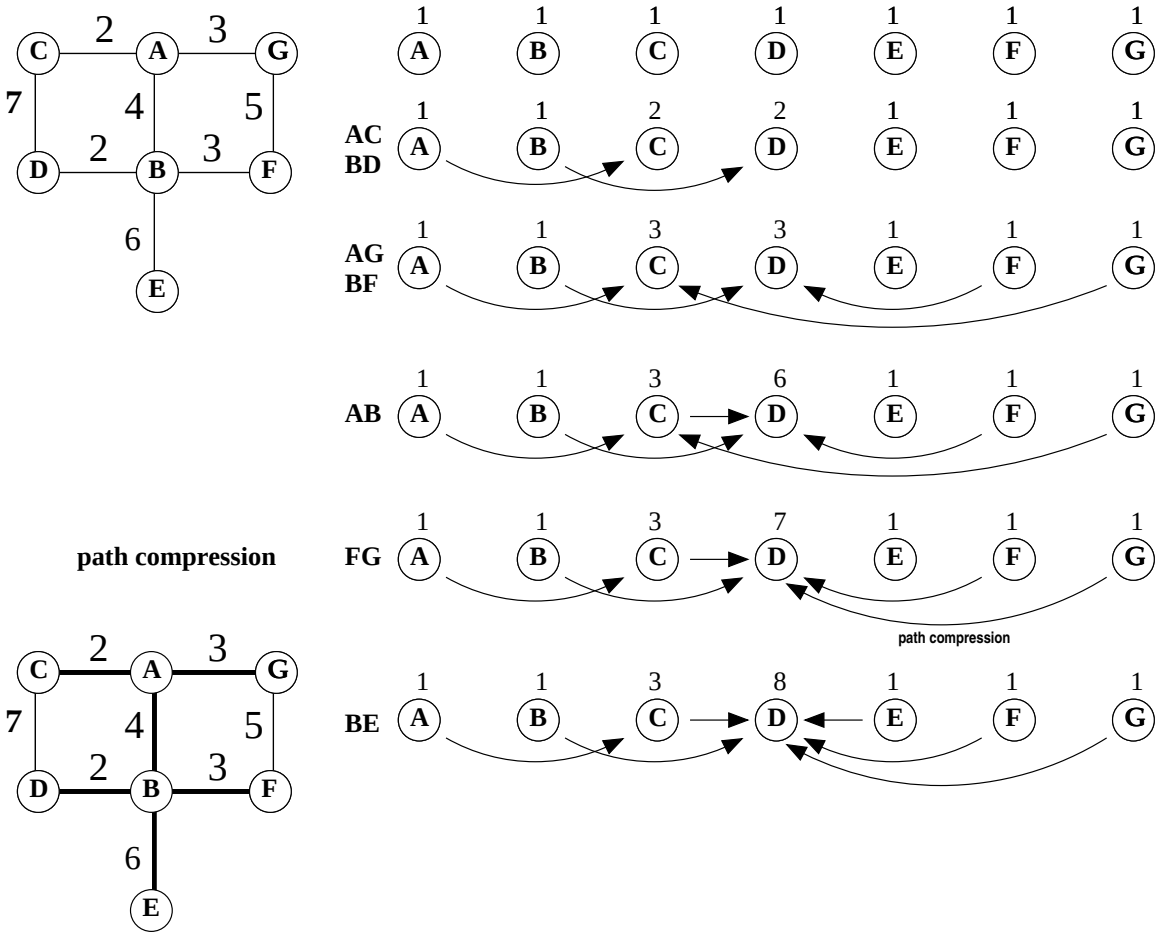
Akra Brazzi method. $\alpha_1 = \alpha_2 = 3$, $\beta_1 = 2/3$, $\beta_2 = 1/3$. $\gamma = 3$

$\sum_{i=1} \alpha_i \beta_i^\gamma = 3(2/3)^3 + 3(1/3)^3 = 3(8/27) + 3(1/27) = (24+3)/27 = 1.$

This is the middle case, so we have

$F(n) = \Theta(n^\gamma \log n) = \Theta(n^3 \log n)$.

7. [20 points] Find a minimal spanning tree of the graph using Kruskal's algorithm and the union/find algorithm. Show the steps of union/find. Indicate any path compression.



8. The following recursive code computes a function `martha(int)`.

```
int martha(int n)
 {
  if(n < 2) return 1;
  else return martha(n/2)+martha((n+1)/2)+n*n;
 }
```

We assume that every arithmetic operation takes one time step.

(i) [5 points] What is the asymptotic value of `martha(n)`?

$\Theta(n^2)$ by the Master Theorem. Asymptotically we write the code as $M(n) = 2M(n/2) + n^2$, then $A = 2$, $B = 2$, $C = 2$. Since $\log_B A = 1 < C$

the answer is $M(n) = \Theta(n^C) = \Theta(n^2)$.

(ii) [5 points] What is the asymptotic time complexity of the recursive code above?

$\Theta(n^1)$ by the Master Theorem. Asymptotically we write the code as
$T(n) = 2T(n/2) + 1$, since it takes only $O(1)$ time to compute n^2. Then $A = 2$, $B = 2$, $C = 0$. Since $\log_B A = 1 > C$
the answer is $T(n) = \Theta(n^{\log_B A}) = \Theta(n)$.

(iii) [5 points] What is the asymtotic complexity of computing `martha(n)` by a standard dynamic programming algorithm?

We compute $M(0), M(1), M(2), \ldots M(n)$ using dynamic programming, and that computation takes $\Theta(n)$ time.

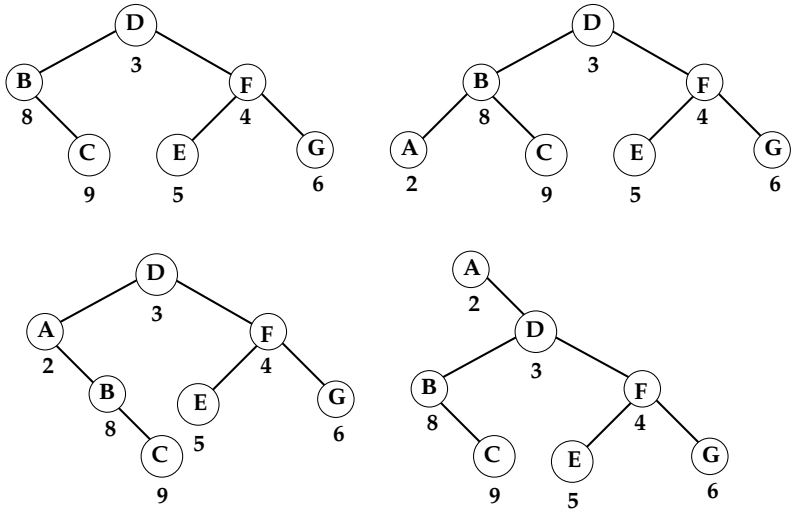
(iv) [10 points] What is the asymptotic time complexity of computing `martha(n)` by memoization, not counting the time to store and find memos?

The time is proportional to the number of memos we need to store. The number of memos is quite small, due to overlapping of the subprograms. It's easier to see that overlapping by looking at an example. Suppose $n = 125$. To compute $M(125)$ we need only $M(62)$ and $M(63)$. To compute those two values, we need only $M(31)$ and $M(32)$. To compute those two values, we need only $M(15)$ and $M(16)$. Continuing, we observe (actually, we could prove) that there are only two values we need at each "level," and there are only $\log_2 n$ levels. Thus the number of entries we need in the search structure is $\Theta(\log n)$, and that is the time complexity of the computation.

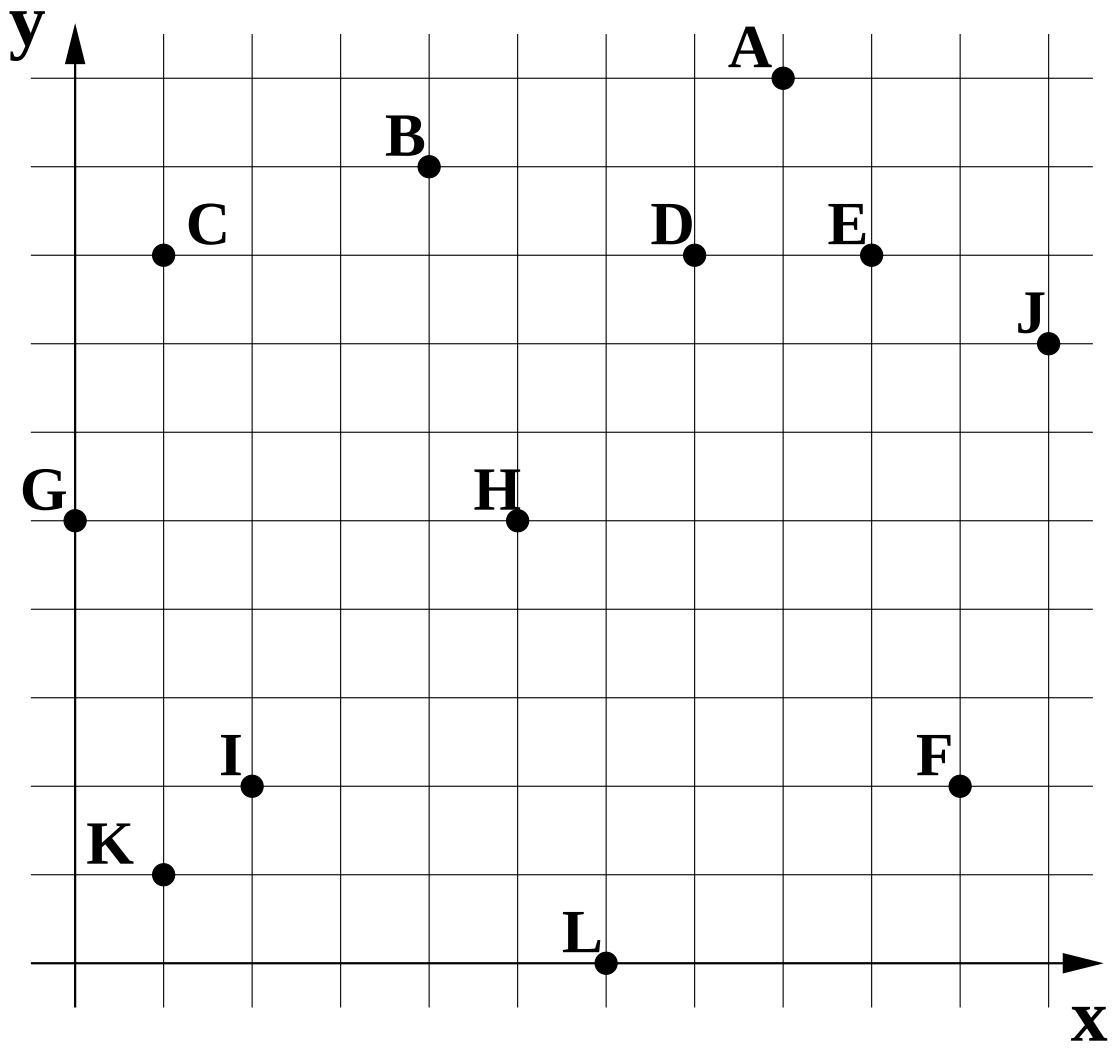
9. [20 points] A treap, with min-heap order, contains the items:
D with priority 3,
B with priority 8,
C with priority 9,
F with priority 4,
E with priority 5,
G with priority 6.

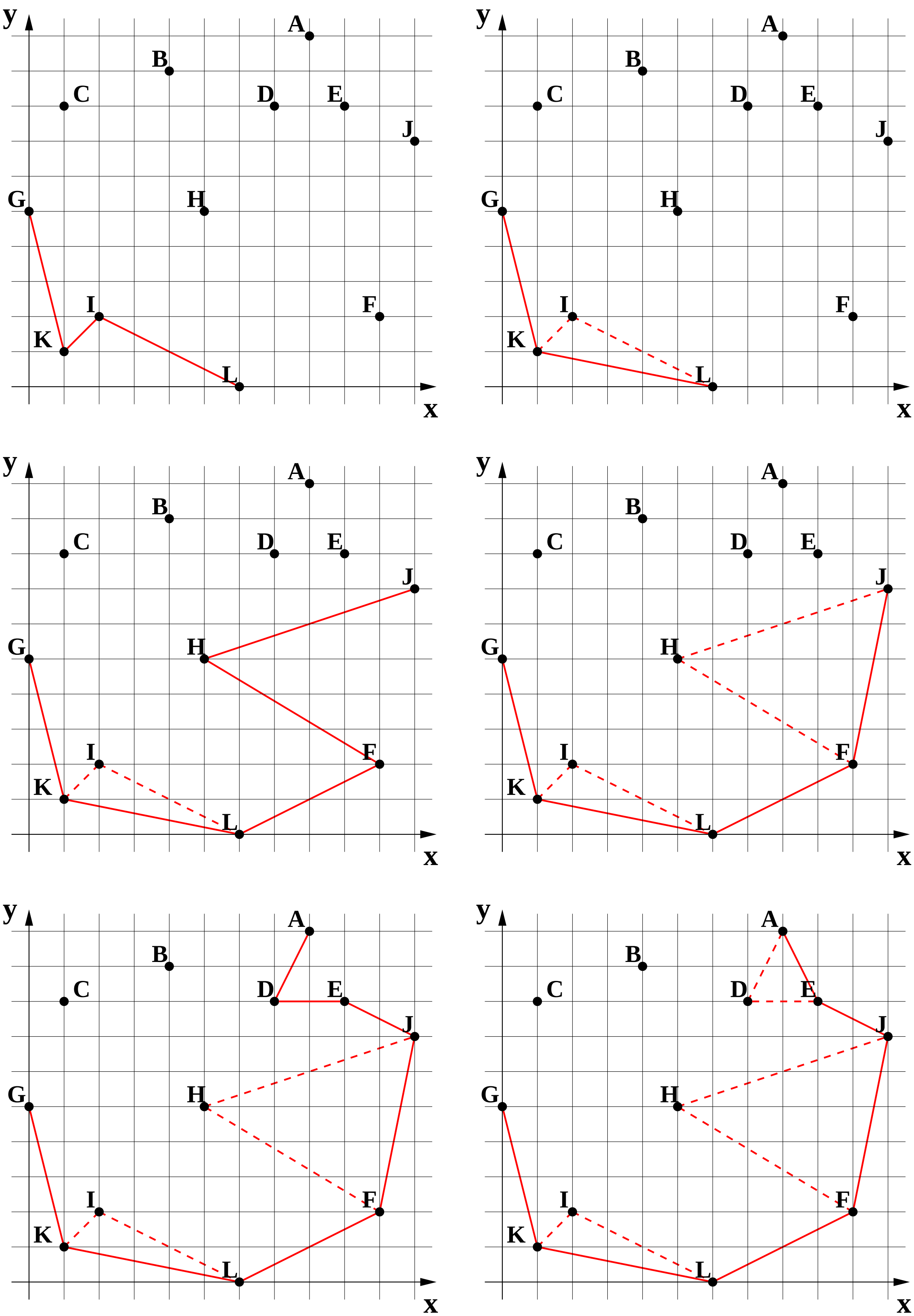
(a) Draw the treap.

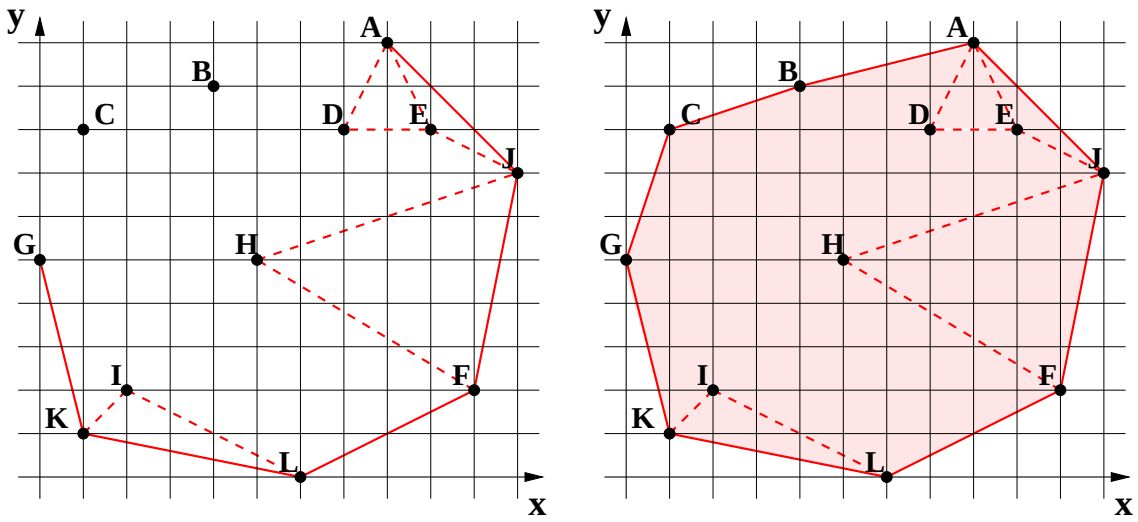
(b) Insert A with priority 2 into the treap. Show the steps, using additional figures.



10. [20 points] Walk through Graham Scan to find the convex hull of the set of points in the plane shown in the figure below. To make the grading task easier, choose G to be the pivot.




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



Our goal is to construct the boundary of the convex hull of the points. We build the boundary as a cycle starting and ending at the pivot, G. We insert the other points in counter-clockwise order of the rays from the pivot to each the other point. This order is: K,I,L,F,H,J,E,D,A,B,C. Each point is added to the boundary once, and is deleted at most once, and thus the algorithm takes $\Theta(n)$ time. We trace this path in red. Whenever we do not make a left turn, we delete the point where we made this turn and the segments connected to it, and connect the points before and after it with a new boundary segment.

We first draw and edge from G to K, then from K to I, then from I to L. There is a right turn at I, hence we delete I and the edges from K to I and from I to L, and create an edge from K to L.

We next draw an edge from L to F, then from F to H, then from H to J. There is a right turn at H, hence we delete H and the edges from F to H and from H to J, and create an edge from F to J.

We draw an edge from J to E, then from E to D, then from D to A. There is a right turn at D, hence we delete D and the edges from E to D and from D to A, and create an edge from E to A. But we now have a right turn at E. So we delete E and the edges from J to E and from E to A, and create an edge from J to A.

We now draw edges from A to B, from B to C, and from C to G, and we are done. The boundary cycle is GKLFJABCG.