```
components = n
for each edge e: marked[e] <- False
repeat
  (x,y) <- random unmarked edge
  marked[ (x,y) ] = True
  rx, ry = find(x), find(y)
  if rx != ry:
    union(x,y)
    components -= 1
until components == 2
V1 <- set of vertices in UFcomponent containing first vertex
V2 <- rest of vertices
return V1,V2
```

1. (i) G is a connected graph with 8 vertices. p is the probability that one run of Karger's randomized min cut algorithm (above) finds a min cut of G. What can you say about p?

(ii) what happens if Karger's algorithm runs on a disconnected graph with 3 or more components?

(iii) modify Karger's algorithm so that it finds a min cut with probability 1 if the graph is disconnected

2. (i) For the sequence below, for each item j in the list, find the length of a longest increasing subsequence that ends with item j.

(ii) Repeat (i) for longest decreasing subsequence.

```
[9, 4, 2, 8, 10, 5, 10, 9, 2, 4]
```