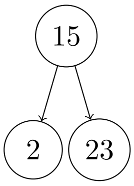
Algorithms Worksheet 5

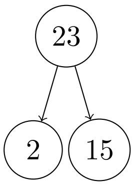
This week there are two question each worth four marks, there are two marks for attendance.

1. Heapify the list $(15, 2, 23, 19, 24, 13, 8)$.

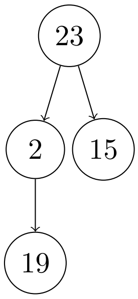
   Solution: The key point is you add each item at the next available slot in the tree and then if it violates the rule that items are smaller than their parent, you swap it upwards until it works. So



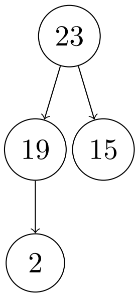
gets changed to



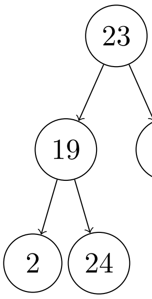
Next adding to the next layer



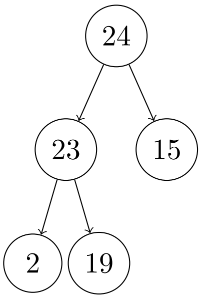
becomes



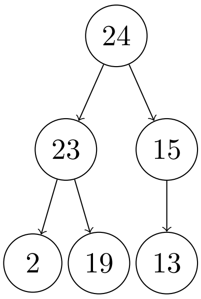
and then



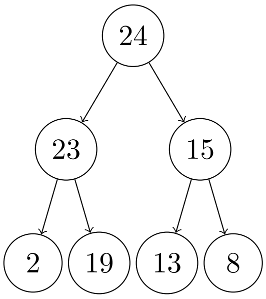
goes to



Next



becomes



and then it finishes.

2. Consider the following example code, taken from the wikipedia article on loop invariants:

```
 1 int max(int n,const int a[]) {
 2         int m = a[0];
 3         int i = 1;
 4         while (i != n) {
 5                 if (m < a[i])
 6                         m = a[i];
 7                 ++i;
 8
 9         }
10         return m;
11 }
```

Explain what is meant by the claim that

> `m` is the maximum of `a[0...i-1]`

is a loop invariant for this function whose purpose is to find the maximum in `a`.

Solution: the statement is true at the start in line 3 because `i=1` and `m=a[0]` which is trivially the maximum; if the statement is true at the start of the loop then the new element is bigger than all the previous ones, in which `m` is set equal the new element, or the maximum remains the same, either way the invariant remains true at the end and if it is true when `i=n` then the algorithm has succeeded.