Assignment 3

Greedy Algorithm

1. Imagine a sequence of containers coming to a dock. When the containers come to the dock, they must immediately be placed on top of a stack. Each container has a letter indicating its size, A being the smallest, Z being the largest. When you place a container on a stack, you can only place it on a container of the same size or larger. Thus, the container with size D can be placed on top of a container with size D or E, but not one of size C. Alternatively, a container can be placed on the ground to create a new stack. Given a sequence of containers that come to dock, write a program to calculate the minimum number of different stacks that have to be created.
   Here is an example input: ABBEFEDBEDFBA
   We have to make 4 stacks for the first 5:

```
     B
A    B    E    F
```

   Now the key decision is, "Where do we put the next item, E? Should we put it on the stack with the top E or top F?" The key to the correct algorithm is realizing that it's better to put the E on the E and not the F. If we put it on the F we get:

```
     B         E
A    B    E    F
```

   then the problem is, if we get an F in the future, then we would be forced to make a new stack. But, if we put it on E, we could handle an F:

```
     B    E
A    B    E    F
```

2. In the art gallery guarding problem we are given a line L that represents a long (straight) hallway in an art gallery. We are also given a set X = {x1, x2, . . . , xn} of real numbers that represent locations where paintings are hung in the hallway. Suppose that a single guard can protect all the paintings within distance at most 1 of his or her position (on both sides). Design an algorithm for finding a placement of the guards that uses the minimum number of guards to guard all the paintings with positions in X.
   a) Assume that we can place a guard at any real number (and they need not be placed at a particular painting, but can be placed in between paintings).
   b) Assume that the guards can only be placed at the same positions of the paintings.

3. Suppose you are given a set S = {a1, a2, …, an} of tasks, where task a_i requires p_i units of processing time to complete, once it has started. You have one computer on which to run these tasks, and the computer can run only one task at a time. Let c_i be the completion time of task a_i , that is, the time at which task a_i completes processing. Your goal is to minimize the average completion time.
   a) Give an algorithm that schedules the tasks so as to minimize the average completion time. Each task must run non-preemptively, that is, once task a_i is started, it must run continuously for p_i units of time. For example, suppose there are two tasks, a1 and a2, with p1 = 3 and p2 = 5, and consider the schedule in which a2 runs first,

followed by a1. Then c2 = 5, c1 = 8, and the average completion time is (5 + 8)/2 = 6.5.

b) Suppose now that the tasks are not all available at once. That is, each task has a release time r_i before which it is not available to be processed. Suppose also that we allow preemption, so that a task can be suspended and restarted at a later time. For example, a task a_i with processing time $p_i = 6$ may start running at time 1 and be preempted at time 4. It can then resume at time 10 but be preempted at time 11 and finally resume at time 13 and complete at time 15. Task a_i has run for a total of 6 time units, but its running time has been divided into three pieces. We say that the completion time of a_i is 15. Give an algorithm that schedules the tasks so as to minimize the average completion time in this new scenario.

4. Suppose that we have a set of activities to schedule among a large number of lecture halls. We wish to schedule all the activities using as few lecture halls as possible. Give an efficient greedy algorithm to minimize the number of lecture halls and determine which activity should use which lecture hall. Assume n is the number of activities to schedule; h is the number of lecture halls available; s_i and f_i are the start time and finish time of some activity a_i.