

### 1. Job Optimization

	Solution	Time Slot 1	Time Slot 2	profit
	1	Job 1	Job 3	55
	2	Job 3	Job 1	55
	3	Job 2	Job 1	65
	4	Job 2	Job 3	60
(a)	5	Job 4	Job 1	70
	6	Job 4	Job 3	65
	7	Job 1	N/A	30
	8	Job 2	N/A	35
	9	Job 3	N/A	25
	10	Job 4	N/A	40

- (b) The optimal schedule has Job 4 in timeslot 1 and Job 1 in timeslot 2 for a profit of \$70.
- (c) A high level greedy algorithm would choose the largest profit with a deadline of 1 or 2, then choose the largest profit with a deadline of 1. In this case, it would choose Job 4, then Job 1.

### 2. Dynamic Programming: Change Making

- (a) The minimum number of coins needed to meet the amount is 3.
- (b) Minimum coin combinations include  $\{1, 2, 5\}$  and  $\{3, 3, 3\}$

(c)	$n$	0	1	2	3	4	5	6	7	8	9
	$f(n)$	0	1	2	1	2	1	2	3	2	3

- (d)
- ```

Change-making( $D[j]$ ,  $n$ ):
     $f[0] = 0$ 
    for  $i = 1$  to  $n$  do
         $temp = \infty$ 
         $j = 1$ 
        while  $j \leq m$  and  $i \geq D[j]$ 
             $temp = \min(f(i - D[j]), temp)$ 
             $j = j + 1$ 
         $f[i] = temp + 1$ 
    return  $f(n)$ 
    
```

### 3. Dynamic Programming: Knapsack Problem

- (a)  
 (b)  
 (c)

#### 4. Greedy Algorithm

(a)

(b)

(c)