

CS375 Small Assignment 9

Due Oct 27th 10PM ET

This small assignment will focus again on the rod cutting problem. We'll modify it again by a little: suppose we are only allowed to make at most k cuts for some integer $k \geq 0$. Develop a $\Theta(nk)$ algorithm to determine the best way to cut a length n rod using **at most** k cuts.

Trace through your algorithm on the input $p = [0, 8, 18, 26, 26, 36, 41, 49, 55, 63, 68, 77, 79, 84]$ and $k = 3$. **Solution:** This time, let's compute $\text{OPT}(j, i)$ = the best value you can get on a length i rod with j cuts. Of course, with 0 cuts we would only get the price of that rod: $\text{OPT}(0, i) = p[i]$ for all i . Additionally, a length 0 rod will get you 0 value no matter how many cuts you're allowed: $\text{OPT}(j, 0) = 0$. If I'm allowed $i \geq 1$ cuts, then the best value I can get is either

- not making a cut, in which case I get $p[i]$ value, or
- make a cut and get the optimal value on whatever's left with one less cut available. If that cut was of length ℓ , this would give value $p[\ell] + \text{OPT}(j - 1, i - \ell)$

This gives us the following recursive formula:

$$\text{OPT}(j, i) = \begin{cases} 0 & \text{if } i = 0 \\ p[i] & \text{if } j = 0 \\ \max(p[i], \max_{0 < \ell < j} p[\ell] + \text{OPT}(i - \ell)) & \text{otherwise} \end{cases}$$

We'll use the following pseudocode to determine the solution dynamically:

```
A = new array of size 0...k x 0...n
```

```
for j = 0 to k:
```

```
    A[j][0] = 0
```

```
for i = 0 to n:
```

```
    A[0][i] = 0
```

```
for i = 1 to n:
```

```
    for j = 1 to k:
```

```
        A[j][i] = max(p[i], max_{0 < \ell < j} (p[\ell] + A[j - 1][i - \ell]))
```

```
j = k
```

```
i = n
```

```
output = empty list
```

```
while i > 0 and j > 0:
```

```
    if A[j][i] == p[i]:
```

```
        add i to output
```

```
        i = 0
```

```
    else:
```

```
        let \ell be such that A[j][i] == A[j - 1][i - \ell] + p[\ell]
```

```
        // note that by construction of A such an \ell must exist
```

```
        add \ell to output
```

```
        i -= \ell
```

```
if i > 0 add i to output # it could be the case that j = 0, ie we ran out of cuts at which
```

```
return output
```

On the given input, we get the following:

i:	0	1	2	3	4	5	6	7	8	9	10	11	12	13
p:	0	8	18	26	26	36	41	49	55	63	68	77	79	84
A[0]:	0	8	18	26	26	36	41	49	55	63	68	77	79	84
A[1]:	0	8	18	26	36	44	52	54	62	67	75	81	89	95
A[2]:	0	8	18	26	36	44	54	62	70	78	80	88	93	101
A[3]:	0	8	18	26	36	44	54	62	72	80	88	96	104	106

Note that

$$A[3][13] = 106 = 88 + 18 = A[2][11] + p[2]$$

$$A[2][11] = 88 = 62 + 26 = A[1][8] + p[3]$$

$$A[1][8] = 62 = 36 + 26 = A[0][5] + p[3]$$

$$A[0][5] = 36 = p[5]$$

which means we output $[2, 3, 3, 5]$ (which makes exactly 3 cuts from a length 13 rod)