P2.

15.1.3

```
Cut-rod-modified (P, n, c)
  if n==0
     return 0
  q=-∞
  for i=1 to n
      q= max(q, P[i] + cut-ROD(P, n-i))
  q=q-c
return q.
```

15.2.4

~~Vert~~ Vertices: $n+(n-1)+(n-2)+\dots 1 = \frac{(n+1)\cdot n}{2}$

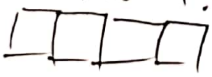
Edges: It has $\sum_{i=1}^{n}\sum_{j=1}^{i}(j-i) = \sum_{i=1}^{n}\sum_{t=0}^{n-i} t$

$$= \sum_{i=1}^{n}\frac{(n-i)}{2}(n-i+1)$$

15.3.5

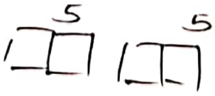
For example, take figure 15.2 as example.

LRod=4.



First cut

5 5



Revenue = 5+5 = 10, which is maximized.

~~So we can tell that.~~ It's impossible because ~~the~~ we can only have ~~1 length, at our first cut. So the previous optimal subproblem doesn't hold.~~

If we have rod=4, we can produce 4 pieces of rods. the original one now to cut it into two which is different from now. So the original solution doesn't hold anymore

15-3

(To be honest again I searched this on google to get hints. It's beyond my ability.

If this is in test I would directly drop dead)

We can equally make this problem into: A and B start from the leftmost point. And travel two different paths and reach the rightmost point

Let $f[i][j]$ stand for the shortest path that A goes to $i$

And B goes to $j$.

when $i = j$

$f[i][j] = \min_{1 \le k \le j-1} (f[k][i] + dis[k][i])$

$i < j-1$

$f[i][j] = f[i][j-1] + dis[j-1][j]$

```
for (j=3  j<=n; j+1 ){
  for (i=1  i<j-1 j-2; i++){
    f[i][j] = f[i][j-1] + dis(j-1, j)
  }
for (k=1 ; k <= j-2 ; k+1          (i=j-1)
    f[j-1][j] = min(f[j-1][j], f[k][j-1] + dis(k,j)),
for (int k=1 ; k <= i-1 ; ++k)     (i=j)
    f[j][j] = min (f[j][j], f[k][j]+dis[k][j]]
```

return $f[n][n]$.

扫描全能王 创建