MaxProfitLimit(m, ℓ): Max profit you can get by cutting a rod of length m into $\leq \ell$ pieces.

$(m \geq 0,\ \ell \geq 1)$

Ultimately, we want to compute MaxProfitLimit(n, k).

MaxProfitLimit(m, ℓ) =

$$\begin{cases} 0 & \text{if } m = 0 \\ p[m] & \text{if } \ell = 1 \\ \max\limits_{1 \leq j \leq m} \left(p[j] + \text{MaxProfitLimit}(m-j, \ell-1) \right) \end{cases}$$

nothing to sell

if $\ell = 1$, can't cut!

o.w., guess length of first piece.

Store answers in $MPL[0..n, 1..k]$.
Deps use smaller m + ℓ,
Solve $m \leftarrow 0$ to n, $\ell \leftarrow 1$ to k.

$O(n)$ time per subproblem, $O(nk)$ problems,

so alg will run in $O(n^2k)$ time.

MaxProfit($P[1..n]$, k):

```
for m ← 0 to n
    for ℓ ← 1 to k
        if m = 0, MPL[m, ℓ] ← 0
        else if ℓ = 1, MPL[m, ℓ] ← P[m]
        else
            most ← -∞
            for j ← 1 to m
                if most < P[j] + MPL[m-j, ℓ-1]
                    most ←            "
return MPL[n, k]
```

3-CGP: Takes undirected $G=(V,E)$.

Decide if we can partition V into three subsets (V_1, V_2, V_3

s.t.

$V_i \cap V_j = \emptyset$ if $i \neq j$

but $V_1 \cup V_2 \cup V_3 = V$)

<u>s.t.</u> the induced subgraph on each subset is complete.

($uv \in E \ \forall \ u, v \in V_i, u \neq v$)

(no restriction on edges between subsets)

Claim: 3-CGP is NP-hard.

Proof via reduction from 3 Color.

Let $G=(V,E)$ be an instance of 3Color.

1) Compute $\bar{G} = (V, \bar{E})$ the complement of G. (i.e. $uv \in E$ iff $uv \notin \bar{E}$)

2) Return answer to 3-CGP on $\bar{G}$.

Takes $O(n^2)$ time.

Must prove G has a proper 3-coloring iff 3-CGP returns Yes for $\bar{G}$.

Suppose G has a proper 3-coloring with colors $1, 2, 3$.

$V_i :=$ all vertices with color i $\forall\, i \in \{1, 2, 3\}$

For any $u \neq v \in V_i$, $uv \notin E$.

So, $uv \in \overline{E}$. So V_i induces a complete subgraph.

Suppose $\overline{G}$ has a 3-CGP $\{V_1, V_2, V_3\}$.

Color each vertex of V_i the color i.

For all $u \neq v \in V_i$ $(\forall\, i \in \{1, 2, 3\})$, $uv \in \overline{E}$. So, $uv \notin E$. So we made a proper 3-coloring. $\square$

Claim: 3-CGP $\in$ NP.

Proof: Given a Yes instance $G=(V,E)$, a good certificate/proof is the parition $V_1 \cup V_2 \cup V_3 = V$.

Can verify each V_i induces a complete subgraph in $O(V^2)$ time.

$\Rightarrow$ 3-CGP is/$\in$ NP-complete.

Reminder: Select$[X[1..m], k]$ returns the kth element of X in sorted order.
Can be run in $O(m)$ time.

rank: the position of an element in sorted order

Given $A[1..n]$ of distinct #s,
$R[1..r]$ of distinct ranks.

Compute element of rank $R[i]$ in A for all $i \in \{1, \ldots, r\}$.

Can solve in $O(nr)$ time by running Select$(A, R[i])$ $\forall i$.

Can you solve it in $O(n \log r)$ time?

Start by sorting $R[i]$ in

$O(r \log r) = O(n \log r)$ time.

Procedure ComputeElements$(A[1..n], R[1..r])$ ← sorted

If $r=0$, return.

Compute element e of rank

$R[\lceil r/2 \rceil]$ by calling

Select$(A[1..n], R[\lceil r/2 \rceil])$

If $r=1$, return.

Partition A around e in $O(n)$ time.

ComputeElements$(A[1..R[\lceil r/2 \rceil]-1],$
$R[1..\lceil r/2 \rceil -1])$

ComputeElements$(A[R[\lceil r/2 \rceil]+1 .. n],$
$R[\lceil r/2 \rceil]+1 .. r])$

after decreasing each by
$R[\lceil r/2 \rceil]$

Time:

Spend $O(n)$ time outside recursion.

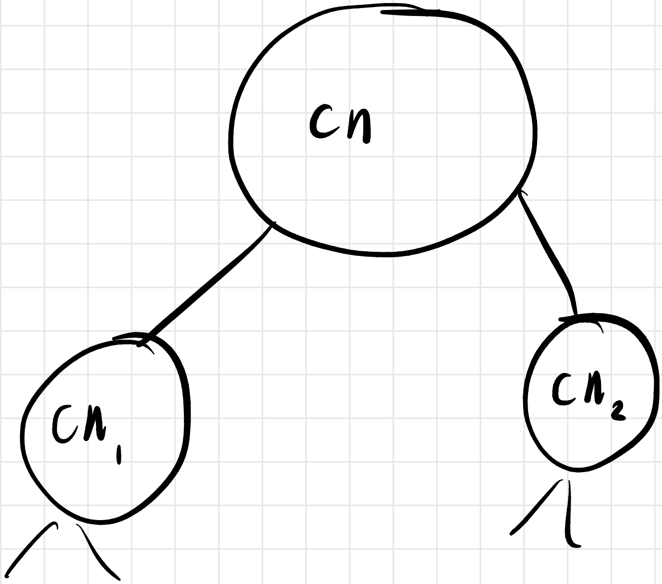
$T(n,r) :=$ running time on $A[1..n], R[1..r]$.

Recursive calls have subarrays of A summing to $n-1$ + rank array about halves.

$$T(n,r) \le T(n_1, r/2) + T(n_2, r/2) + O(n) \quad (n_1 + n_2 = n-1)$$

worst

recursion tree:



$\Sigma = cn$

$\Sigma \le cn$

$\Sigma \le cn$

$\vdots$

$\lg r$ levels, so total time is $O(n \log r)$.

Proof of correctness: If $r=0$ or $r=1$, correctness is immediate thanks to base cases.

O.W, $r > 1$.

We do compute element of rank $R[\lceil r/2 \rceil]$.

Elements of ranks $R[1 .. \lceil r/2 \rceil - 1]$ come earlier in sorted order it suffices to search ranks 1 to $R[\lceil r/2 \rceil] - 1$, which we do by induction on n.

Other elements have higher rank, so it suffices to search in ranks $R[\lceil r/2 \rceil] + 1 .. n$, but note relative to subset those elements have rank decreased by

$R[\lceil r/2 \rceil]$. We find them recursively by induction on n, also.