16.1.5 P3 LinXi Li 1

~~15.1.4~~

~~GREEDY-Activity-selector-modified (S, V)~~

n = ~~s.length;~~

~~A={a1}~~ (the a_1 is the first event in the sequence sorted by finishing time

~~Sort (A, V) ← sort the sequence a by value~~

$V_0 = 0$

~~for m = 2 to n:~~

~~If s[m] ≥ f[k]:~~

~~A = A ∪ {a_m}~~

~~V = V_0 + V[a_m]~~

~~k = m~~

GREEDY-Recursive-activity-selector (modified) (s, f, k, n)

m = k+1 ~~V_0 = 0~~ V = sort[V_0] ← sort Value from largest to smallest

while m ≤ n and s[m] < f[k] ~~and V[a_m]]~~

~~V_0 + = V[a_m]] m = m+1~~

If m ≤ n

return {V_m} ∪ Recursive-Acivity-selector (s, f, m, n)

else return 0

we simply go over the sorted, values from largest to smallest, and If two Points, are compatible, we add ~~them~~ together in the array. then the output array would be largest.

16.1 (Inspired by solutions on CSDN)

a. we can determine the largest coin, whose value is less or equal to n. Consider this value as ~~m~~ c. the recursively solve (n−c), do same thing to value (n−c)

for 1 ≤ n ≤ 5, c = 1. then the solution consists only with Pennies. The assumption holds

for 5 ≤ n ≤ 10, c = 1. If we minus 5, then subproblem would be 1 ≤ n−c ≤ 5. As the first statement, the equation hold.

for $10 \le n \le 25$, we simply minus with 10, then ~~the~~ the subproblem would be $5 \le n-c \le 10$. As the statement 2 indicates, the assumption holds.

for $25 \le n$, as before, subproblem would be $10 \le n-c \le 25$, as the statement states, the equation holds.

b. ~~for $i=0,1,\dots,k-1$, c^i in optimal solution used is less than c~~
we can ~~improve the solution by using~~

for example, suppose the largest coin $c^j \le n$, then we solve the subproblem of $n-c^j$ cents

for non-greedy solution, the result is $\sum_{i=0}^{j-1} a_i c^i = n \ge c^j$

the sum of $\sum_{i=0}^{j-1} a_i$ would be result.

~~As~~ $a_i \le c-1$, so $\sum_{i=0}^{j-1} a^i c^i \le \sum_{i=0}^{j-1} (c-1)^i c^i$

It's contradiction, so optimal solution is the best solution

$$= (c-1)\sum_{i=0}^{j-1} c^i = (c-1)\frac{(c^j-1)}{(c-1)}$$

$$= c^j - 1 < c^j$$

$C = \{6, 5, 1\}$ $n=15$, greedy algorithm returns with count $= 5$. Actually count only needs to equal to 3.

d. Let $C[j]$ denote the minimum coins we need to make change for j cents

If $j=0$:
$C[j]=0$
else:
$1+\min_{1\le i\le k}\{C[j-d[i]]\}$

It ~~~~ runs in $O(nk)$ time.