0-1 Knapsack

Wednesday, April 4, 2018 6:52 PM

0 don't take value

1 take the value

Can be solved by dynamic programming

what is my state?

LCS 2 integers

location in respective strings.

Change 2 integers

target

smallest

Floyd's 2 integer

starting index

ending index

Items have a weight w a value v

and our bag has capacity C.

maximize sum of values of the selected items such

the capacity is greater than or equal

to the sum of the weights of the selected

items

$a_i \in \{0,1\}$

$$\text{maximize} \sum_{i \in I} a_i \cdot v_i$$

s.t.

$$C \geq \sum_{i \in I} a_i \cdot w_i$$

Decide at each point while receiving items in an order whether we take the item or not.

inefficient

```
function (C , S, index ){
    if( memo [C][S][index] ≠ SENT ){
        return memo [C][S][index];
    }

    if( C < 0)
        return -∞;
    if(index = T.length )
        return S;
    int first = ((C-index.w, S+index.v, index+1) // take
    second = ((C, S, index+1); // don't
    memo = max (first, second);
    return memo;
}
```

```
function (C ,     index ){
    if( memo [C]    [index] ≠ SENT ){
        return memo [C]    [index];
    }

    if( C < 0)
        return -∞;
    if(index = T.length )
        return 0;
    int first = ((C-index.w, index+1) + index.v // take
    second = (C, index+1) ;          // don't

    memo = max (first, second);

    return memo;
}
```