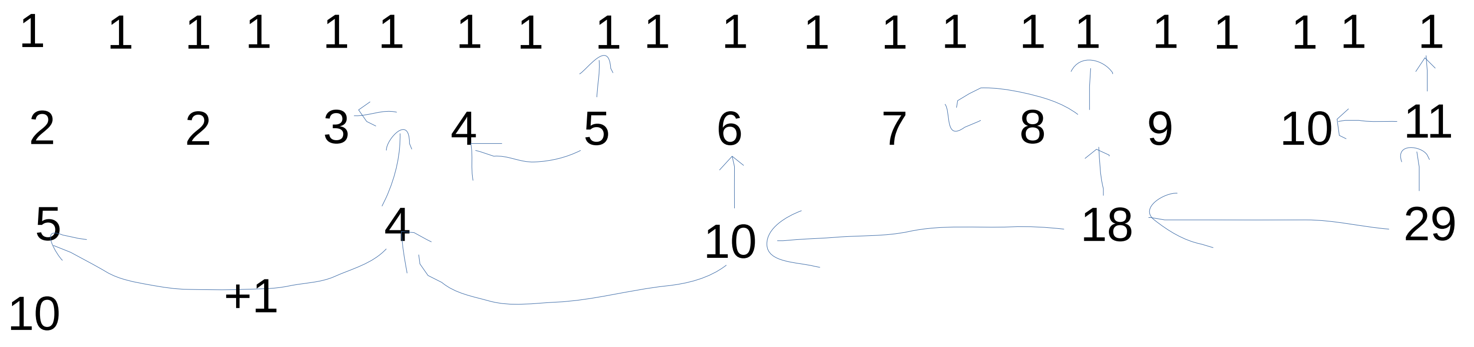
1. Idea:
-Tree
- depth first
- possible later optimization with paralell divide and conquer
- if legal leaf is reached → add to set   Hashset<(u32 x 8)>
- use structure here to aggregate the different coins to a bag

2. Idea

Dinamic programming:
- from size 1 to n
- using the old solutions when possible → Hashmap with <left_size, possible_combiations_to_fill>
- else expand like a node
- use iteration instead of recursion:
- use work stack
    - if empty push 1 to n on it
    -

!! the problem here is that that the DIFFERENT ways are on intrest



20*1
18*1+1*2
..
2*1 + 8*2
10*2

== 11

4*5
3*5 + {1*2+3*1, 2*2+1*1, 5*1} = 3
2*5 + {10*1,… 5*2} = 6
1*5 + {15*1+0*2, 1*1+ 7*2} =8