Mini HW #6

資工二 陳盈如 B05902118

November 6, 2017

1.

Algorithm 1: Sum

Input: s_i, sum
Output: $ans[], ans\_len$

```
for i = 0; i < len; i++ do
    if s_i ≤ sum then
        upper_bound = i;
for i = upper_bound; i ≥ 0; i-- do
    if sum ≥ s_i then
        sum -= s_i;
        ans[ans_len] = s_i;
        ans_len++;
if up > 0 then
    return False;
else
    return ans[], ans_len;
```

First of all, go through the array and find the biggest number in the array which is smaller than *sum*. Seeing that it's an superincreasing sequence, the sequence goes from small to big. When we find a value bigger than the sum, we can break out of the loop.

Then, use *sum* to minus the number in the array from the biggest number we just get to the smallest number. If *sum* becomes smaller than the number in the array, jump to the next value in the array. Finishing doing what mentioned above, if sum is zero, return. If not, the question is impossible to find an answer.

2.

- Greedy Choice Property:
  - 選擇陣列裡小於 sum 的最大值並減去它(假設此值為 x)，也就是說 Optimal Solution 裡面有這個選擇，此選擇為 Greedy Choice。
  - 利用反證法：假設不選擇 Greedy Choice，有更好的解。
  - 因為題目為 superincreasing sequence，若不選擇此選項，即使選擇了前面所有的數字，數字總合也不可能是 sum。假設：

    $S[] = \{1, 2, ..., z, x, y, ...\}$

    $x < sum < y$

    $\because 1 + 2 + ... + z < x$

    $\therefore 1 + 2 + ... + z < sum$
  - 沒有更佳解，出現矛盾！
- Optimal Substructure:
  - Optimal Solution = Greedy Choice + Optimal Solution to subproblem，每選擇一個 Greedy Choice，選過的數字就不用管他，從剩下的數字再繼續做 Greedy Choice，所以所有 Greedy Choice 的集合就是最後答案。