1 Backtracking

1.1 Text Segmentation

From page 83-84:

$$Splittable(i) = \begin{cases} \textsc{True} & \text{if } i > n \\ \bigvee_{j=i}^{n} \left(\textsc{IsWord}(i,j) \wedge Splittable(j+1) \right) & \text{otherwise} \end{cases}$$

```
⟨⟨Is the suffix A[i..n] Splittable?⟩⟩
SPLITTABLE(i):
  if i > n
      return TRUE
  for j ← i to n
      if ISWORD(i, j)
          if SPLITTABLE(j + 1)
              return TRUE
  return FALSE
```

Suppose IsWord is defined as:

$$\textsc{IsWord}(i,j) := (A[i..j] = \text{reverse}(A[i..j]))$$

- Trace $Splittable(1)$ on the array, $A[1..9] =$ `"AABCCBAAZ"`.

- Suppose we want to know the *minimum number of splits* needs to split $A[1..n]$ into valid "words" (according to IsWord). Let $MinSplits(i)$ be a function that computes the minimal number of splits needed to split $A[i..n]$ into valid "words". Define $MinSplits(i)$.

2 Subset Sum

The SubsetSum problem: Given an array $A[1..n]$ of positive integers and a *target* integer T, is there a subset of the elements of A that add up to T?

5	6	3	6	5	2	8	10

- $T = 30$?
- $T = 100$?
- Write an English description of what you want to calculate. (Work backwards from $A[n]$. Decide whether to include it or not, ...)
- Write a recursive definition of SubsetSum().
- Give a couple sentences (in English) of why your recurrence should work.

3 Longest Increasing Subsequence

Given an integer array $A[1..n]$, compute the length of the longest possible sequence of indices (not necessarily contiguous), $1 \leq i_1 < i_2 < \ldots < i_l \leq n$, such that $A[i_k] < A[i_{k+1}]$ for all $1 \leq k < l$.

Example:

5	-6	3	6	-5	2	8	10

- What is the optimal value for the array above?

- Develop a recursive algorithm going left-to-right through the array, thinking "do I include this element or not?" for each element. (Will need to keep track of previously largest element.)

- Time complexity?

Alternate way of thinking about LIS: Ask "What's the longest subsequence starting from me?'.

- Let $LISStart(i)$ be the length of the longest increasing subsequence among indices $i..n$ that starts at index i. Develop a recurrence.