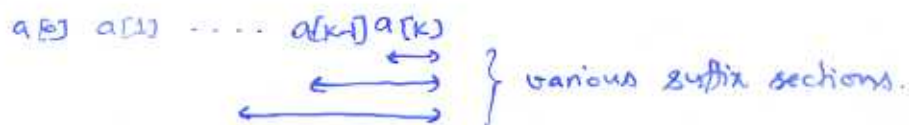


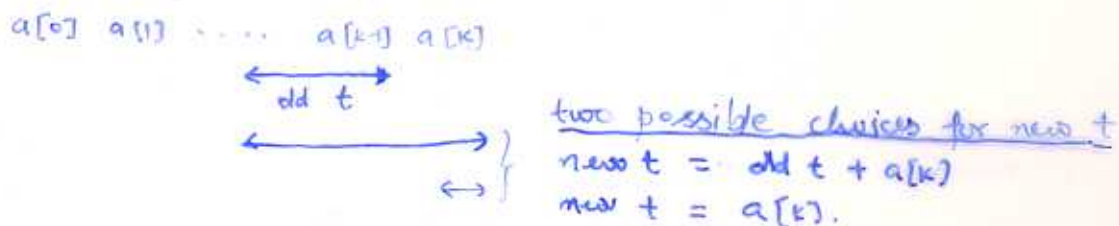
Example: Minimal-sum Section of Array

- Given an array of size n , find a minimal sum over all subsections.
- array of length $n > 0$: $a[0] a[1] \dots a[n-1]$
- sum of subsection from i to j : $S_{ij} := \sum_{k=i}^j a[k]$
- Goal is to compute $s = \min_{0 \leq i \leq j < n} S_{ij}$

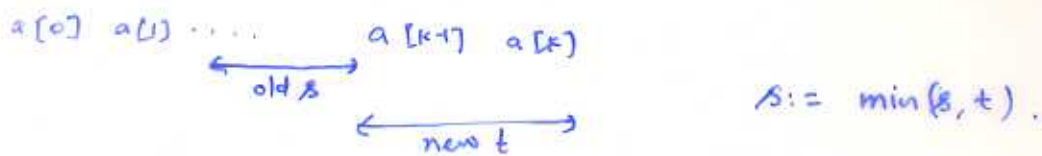
Example: $[-1 \ 3 \ -2]$ (consider different subsections and compute sums)
 For this we make a single pass of the array storing two values t & s .
 t stores minimal sum of all "suffix sections" so far.



t is updated as follows: $t := \min(t + a[k], a[k])$



s stores minimal sum of all "sections" so far, and is updated as follows



So we obtain following program:

```

k := 0;
t := a[0];
s := a[0];
while (k < n) do {
  t := min(t + a[k], a[k]);
  s := min(s, t);
  k := k + 1;
}
  
```