Basics of Algorithm

September 4, 2024 12:17 PM

Algorithm is an ordered sequence of steps that transfo

Example

Sum an array of Integers.

Input: An array A[1...n] of n integers.

Output: Sum of elements of A.

Pseudocode

Cost of Execution

```
Sum ← 0                          ——— c1
for ( i = 1; i ≤ n ; i++)        ——— c2
    Sum ← Sum + A[i]             ——— c3
return sum                       ——— c4
```

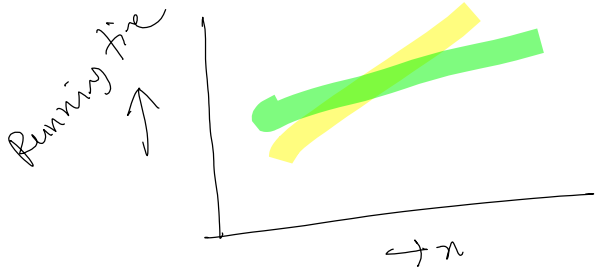
Total cost = $c_1 + n c_2 + n c_3 + c_4 =$

c_1 ↓ 2, c_2 ↓ 4, c_3 ↓ 6, c_4 ↓ 2

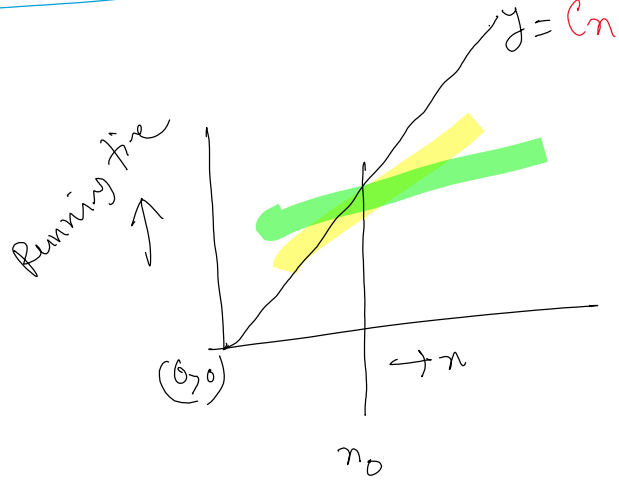
Example for Sam's Laptop

Assume that for Lily's laptop we have 5

These both look like linear functions



I am claiming there are constants C and n
such that the running time for both these lab
are at most C*n for every $n \geq n_0$



Lily
$50 + 7n < 100 + 100n$

$\leq 200n$

Sam
$4 + 10n$

If I choose $C = 200$ and $n_0 = 1$

then for any $n \geq n_0$ cn or $200n$ wi
than Sam or Lily's running time.

Formally we say that the asymptotic running time of the algorithm is $O(n)$.

Claim: The algorithm returns correct sum - for A[1

proof: Base Case: when $i = 1$ then sum ← 0+A[

Ind. Hypo: For every $i < n$ the algo will

For $i = n$

by induction hypothesis at the $(n-1)^{th}$

sum has the correct sum of A[1.

and at the n^{th} iteration we get sum ←

Hence sum will return the correct sum

If you are not careful, you can make easy mistak

Mr. Minion went to a shop and saw a blue

Base Case: I saw a banana and that is b

Ind. hyp: Assume that a set $S = \{b_1, b_2 \cdots, b_k\}$ of $\leq k$ bannan

Consider a set of $(k+1)$ bananas $\{b_1 \cdots b_k, b_{k+1}$

, ..., blue bananas.

prove this set only contains

Mr. minion removed one banana b_1 from

get $S_1 = \{ b_2 \cdots b_{k+1} \}$

Mr. Minin removed b_{k+1} to get another

$S_2 = \{ b_1 \cdots b_k \}$.

By ind. hyp. S_1 and S_2 only contains

Therefore $S_1 \cup S_2 = S$ only contains