Friday, September 12, 2014 1:24 PM

Array
List
Stack
Queue

Abstract
Data
Type (ADT)

ADT Stack (stack object)

Data Domain

S – set of all possible stacks of items drawn from set I

I – set of all possible stack items

B – Boolean {0,1}

Operations

Push: $S \times I \rightarrow S$

Pop: $S \rightarrow S \cup \overset{message}{M}$

Top: $S \rightarrow I \cup M$

Empty: $S \rightarrow B$

Create: $\emptyset \rightarrow S$

$i \in I \quad s \in S$

top(push(s, i)) ::= i

pop(push(s,i)) ::= s

empty(push(s,i)) ::= 0

empty(create()) ::= 1

top(create()) ::= $\boxed{?}$

pop (create()) ::= $\boxed{?}$

s = create()

s = push(s, i)

push(pop(s), i2) ? yes

push(pop(pop(s))) ? no

message

Implementation

Efficiency

space vs time

Requirement { customer

↓

Design Specification

Analysis of Algorithms

Time efficiency:

Best
Average
Worst } behavior given some arrangement of data

$$1 + 2 + 3 + --- + N$$

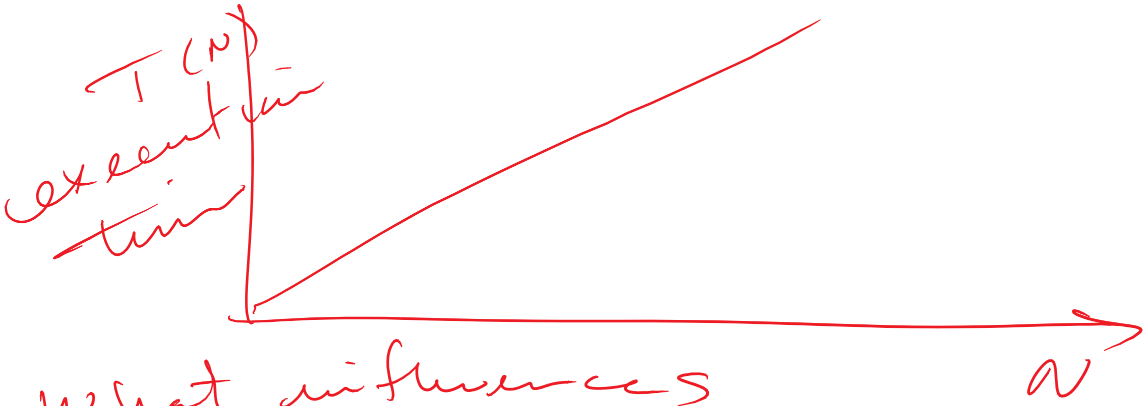
time ins ~~#~~ times executed

```
          int sum ( int N)
          {   ~~int total;~~
 A    1       total = 0;
              ~~int i;~~
 A    1       i = 1;
 B    N       while ( i <= N)
              {     total = total + i;
 C    N               i = i + 1;
 C    N       }
 D    1       return total;
          }
```

$$T(N) = \overset{2}{\cancel{2}A} + \overset{1 \cdot N}{B \cdot N} + \overset{2N}{2C \cdot N} + \overset{1}{D}$$

$$A = B = C = D = 1$$

$$\boxed{T(\underline{N}) = 3N + 3}$$



what influences execution time

- size of input (N)
- arrangement of data
- ~~language / implementation~~ detail

- algorithm

$1 + 2 + 3 \ldots N =$

$N * (N+1) / 2$

Times

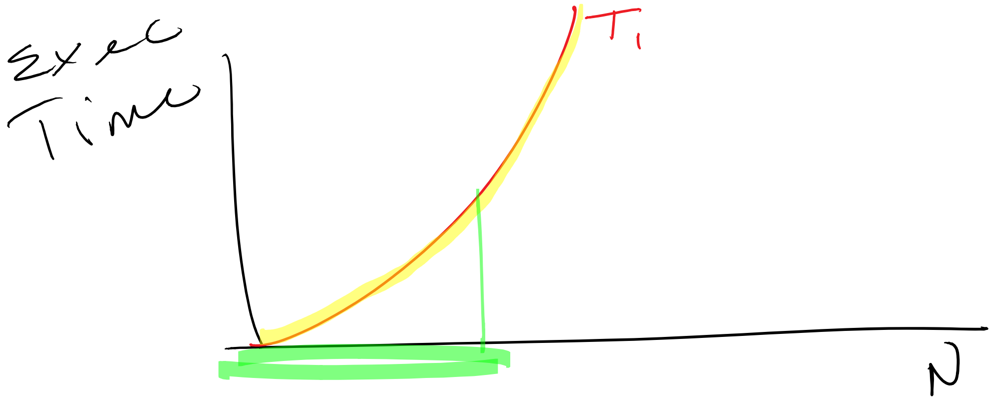
```
int sum ( int n )
{
1       return n*(N+1)/2;
}
```

$$\boxed{T(N) = 1}$$

$N = \infty$

$T_1(N) = 5N^3 + 2N^2 + 6N + 8$

$T_2(N) = 4N^3 + 5N^2 + 27N - 5$


Exec Time
T1
N