Asymptotic Analysis

Worst Case vs Expected Case

Abstract measures of performance will still depend on actual input data

eg. Exaustive Sequential Search

```
public int eSearch(...){
    ...
    i = 0;
    while (a[i] != goal && i < n){
        i++;
    }
    if (i==n){      //Goal not found
        return -1;
    }
    else{
        return i;
    }
}
```

Abstract time

- Goal is first element in array - a units
- Goal is the last element in the array - $a + bn$ units;

for some constants a nd b

Worst Case scenario

Choose which data has the worst time/space requirements. In the case of eSearch, the worst cast complexity is $a+bn$

Advantages

- Reletivly Simple
- Gives upperbound or garrentee of behaviour/can't preform worse

Disadvantages

- Worst case could be unrepresentative of realistic results
- Can't get anyone to buy it

Since we want behavior garrentees we will usually consider worst case scenario

(note there is also best case scenario such as car salesmen or stock brokers)

Expected Case Analysis

What happens on the average or expected case. For `eSearch`, $a+$$bn \over 2$ assuming a uniform distribution of input.

Advantages

- More realistic indicator
- Reduces affects of outlier samples eg. `QuickSort` is usually the fastest even though it has worst time complexity.

Disadvantages

- Only possible if we know the distribution over examples
- More difficult to calculate
- Often does not primovide significanty more information than worst case scenario
- May be misleading

Asymptotic Growth Rates

Comparing data structures with an analytical approach. Focus on:

- independant of run time enviroment
- improves understanding of data structures

We are interested in comparisons in terms of *growth rates*.

Theortical analysis permits us to do a deeper comparison.

We wish to be able to make statements such as:

- Searching for a given element in a block of n distinct elements takes n comparisons in the worst case.
- Searching for a given element in a ordered list takes at least $log(n)$ comparisons in the worst case.

These are lower bounds in the worst case, they tell us we are never going to do any better regardless of the algorithm.

They reflect growth rates.

Why Asymptopia

We would like to have a simple description of behaviour for use in comparison.

- Evaluation can be misleading consider the functions $$t_1 = 0.002m^2, t_2 = 0.2m, t_3 = 2log(m)$$
- Want the *closed form.* eg. $$n(n+1) \over 2$$ not $n+(n-1)+...+2+1$
- Want simplicity.

What simply function does it behave *like*?

Solution

Investigate what simple function the more complex one tends to or the asymp- totically approaches as the argument approaches infinity, ie in the limit.

eg. if we wanted to approximate $n^4 + n^2$ by n^4

How much error?

n	**n^4**	**n^2**	**$n^2 \over n^4+n^2$**
1	1	1	50%
2	16	4	20%
5	625	25	3.8%
10	10000	100	1%
20	16000	400	0.25%
50	6250000	2500	0.04%