UCSC Silicon Valley Extension

Advanced C Programming

Analysis and Design of Algorithms - part 2

Instructor: Radhika Grover

Overview

- Probability Theory
  - Events, outcomes, sample space, probability distribution
- Average running time for merge sort
  - Example
  - Analysis

Probability theory : review

- Outcome - Result of a random experiment

Example : Tossing a single dice (6 outcomes)

Flipping a coin once (2 outcomes)

Flipping a coin twice (4 outcomes)

Probability of any outcome = $\dfrac{Number\ of\ ways\ it\ can\ occur}{Total\ number\ of\ outcomes} = \dfrac{1}{6}$

Event

- Event includes one or more outcomes

Example : Rolling an even number with a single dice [2, 4, 6]

Getting exactly one head when a coin is tossed twice

(HT, TH)

Probability theory : sample space

Set of all possible outcomes

Example : Tossing a single dice = {1, 2, 3, 4, 5, 6}

Flipping a coin once = {H, T}

Flipping a coin twice = {HT, HH, TH, TT}

Probability theory : probability distribution

Links each outcome with its probability of occurrence

Example : Flipping coin twice

Number of heads	**Probability**
0	$\frac{1}{4}$
1	$\frac{2}{4}$
2	$\frac{1}{4}$

Cumulative probability : Probability that the value of a random variable falls within a given range

$P(X \leq 1)$ = probability of getting one or zero heads = $P(X=0) + P(X=1) = \frac{3}{4}$

Probability theory : expected value

X = numerically - valued discrete random variable

S = sample space = distribution function

E(X) = expected value of *X* = mean or average of *x*

$$= \sum_{x \,\epsilon\, S} x \; P_r(X = x)$$

Probability theory

Toss a fair coin two times - *x* is the number of heads that appear

$$E(x) = \sum_{x \ \epsilon \ 0,\ 1,\ 2} x \ P_r(X = x)$$

$$= x \ P_r(X = 0) + x \ P_r(X = 1) + x \ P_r(X = 2)$$

$$= 0 * \frac{1}{4} + 1 * \frac{1}{2} + 2 * \frac{1}{4}$$

$$= 1$$

Probability theory : example

Example of runs for a coin toss : $\underbrace{H\ H}_{\text{Run1}}\ \underbrace{T\ T}_{\text{Run2}}\ \underbrace{H}_{\text{Run3}}\ \underbrace{T\ T\ T\ T}_{\text{Run4}}$

A fair coin is tossed 3 times, find the expected number of runs

Sample space : x = 1 run => $P(X) = \frac{2}{8}$

x = 2 runs => $P(X) = \frac{4}{8}$

x = 3 runs => $P(X) = \frac{2}{8}$

$$E(X) = 1 * \frac{2}{8} + 2 * \frac{4}{8} + 3 * \frac{2}{8}$$
$$= \frac{1}{4} + 1 + \frac{3}{4} = 2$$

H H H	0 0 0
H H T	0 0 1
H T H	0 1 0
H T T	0 1 1
T H H	1 0 0
T H T	1 0 1
T T H	1 1 0
T T T	1 1 1

Average running time

- Find the expected running time of algorithm *E(t)*
- *S* is the sample space of all inputs
- *X* is an input to the algorithm that $X \in S$
- *t(X)* = time taken by algorithm on input *X*
- *P(X)* is the probability distribution of *X*

$$E(t) = \sum_{x \, \epsilon \, S} t(x) * \; P_r(X = x)$$

Average running time

- Find the average number of comparisons during merge
- Merge two random sorted sublists
- Number of comparisons $C = p + q - s$

  p and q : length of sublists

  s = number of elements remaining in one sublist when the other sublist becomes empty (i.e number of largest elements in one subfile)

Find ***E(C)***

Average running time - example

Sublist 2 contains 4 largest elements

$s = 4$

$p = 3$: 1, 2, 3

$q = 4$: 4, 5, 6, 8

Number of comparisons = 3

Number of combinations = 1

Average running time - example

Sublist 2 contains 3 largest elements

$s = 3$

$p = 3$	1, 2, 4	1, 3, 4	2, 3, 4
$q = 4$	3, 5, 6, 8	2, 5, 6, 8	1, 5, 6, 8

Number of ways we can select 1 element from 1, 2, 3 = ${}^{3}C_{1}$

Number of comparisons = 4 (The general formula is $p+q-s$)

Number of combinations = 3

Average running time - example

Sublist 2 contains 2 largest elements

s = 2

	1	1	3	2	2	1
p = 3	2	3	4	3	4	4
	5	5	5	5	5	5
	3	2	1	1	1	2
q = 4	4	4	2	4	3	3
	6	6	6	6	6	6
	8	8	8	8	8	8

Number of ways we can select 2 elements from 1 to 4 = $^{4}C_{2}$

Number of comparisons = *p+q-s* = *5*

Number of combinations = $^{p+q-3}C_{q-2}$ = *6*

Average running time - example

Sublist 2 contains 1 largest element

s = 1

p = 3

1	1	1	1	2	2	2	3	3	4
2	3	4	5	3	4	5	4	5	5
6	6	6	6	6	6	6	6	6	6

q = 4

3	2	2	2	1	1	1	1	1	1
4	4	3	3	4	3	3	2	2	2
5	5	5	4	5	5	4	5	4	3
8	8	8	8	8	8	8	8	8	8

Number of ways we can select 3 elements from 1 to 5 = $^5C_3$

Number of comparisons = *p*+*q*-*s* = *6*

Number of combinations = 10

Average running time - example

Sublist 1 contains 3 largest elements

$s = 3$

$p = 3$

5
6
8

$q = 4$

1
2
3
4

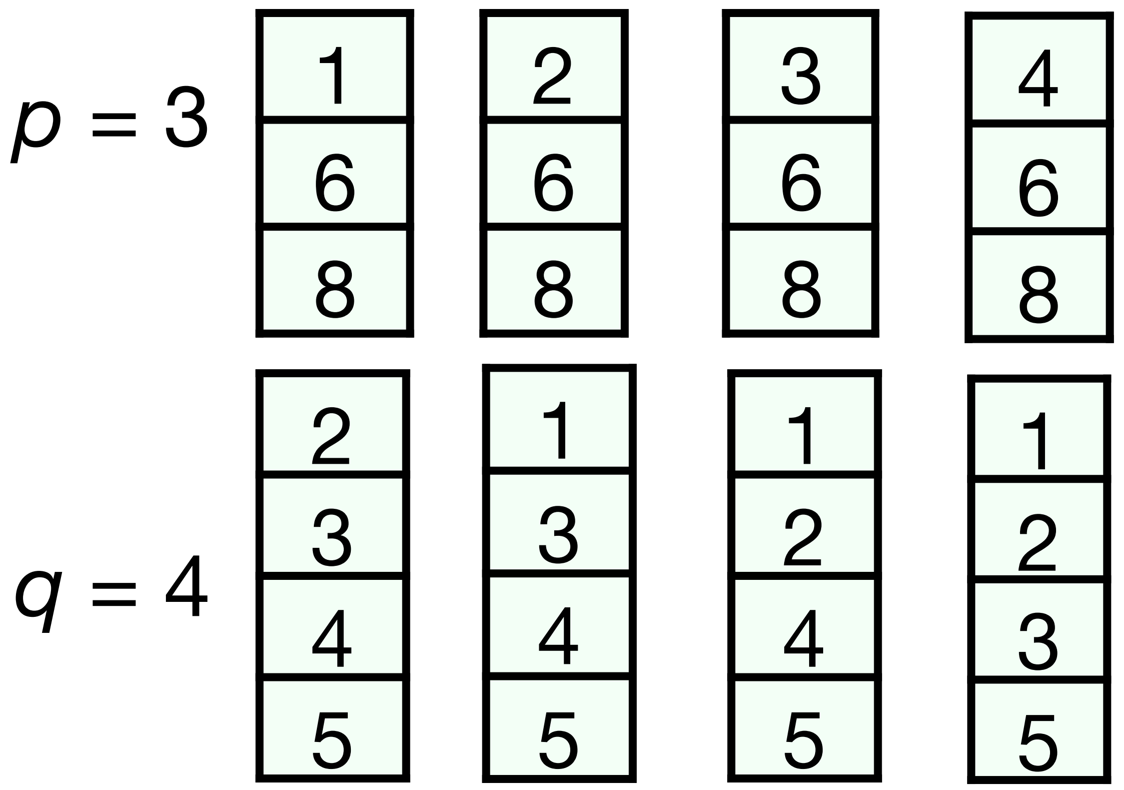
Number of comparisons = $p+q-s$ = 7-3 =4

Number of combinations = ${}^{3}C_{0}$ = 1

Average running time - example

Sublist 1 contains 2 largest elements

$s = 3$



Number of comparisons = 5

Number of combinations = 4

Average running time - example

Sublist 2 contains 2 largest elements

$s = 2$

$p = 3$	1, 2, 8	1, 3, 8	1, 4, 8	1, 5, 8	2, 3, 8	2, 4, 8	2, 5, 8	3, 4, 8	3, 5, 8	4, 5, 8
$q = 4$	3, 4, 5, 6	2, 4, 5, 6	2, 3, 5, 6	2, 3, 4, 6	1, 4, 5, 6	1, 3, 5, 6	1, 3, 4, 6	1, 2, 5, 6	1, 2, 4, 6	1, 2, 3, 6

Number of comparisons = 6

Number of combinations = 10

Average running time -analysis

c = number of comparisons for sublists with length p and q

$E(C)$ = Expected value of C $= \sum_{s \geq 1} c \, P_r(C)$

$$= \frac{3*1+4*3+5*6+6*10+4*1+5*4+6*10}{1+3+6+10+1+4+10}$$

$$= \frac{189}{35} = 5.4$$

Average running time -analysis

$$E(C) = \sum_{s \geq 1} c \; P_r(C = c)$$

where c = number of comparisons for a given value s

$$= \frac{\sum_{s=1}^{p}(p+q-s)\,{}^{p+q-(s+1)}C_{p-s} + \sum_{s=1}^{q}(p+q-s)\,{}^{p+q-(s+1)}C_{q-s}}{\sum_{s=1}^{p}\,{}^{p+q-(s+1)}C_{p-s} + \sum_{s=1}^{q}\,{}^{p+q-(s+1)}C_{q-s}}$$

$$= \frac{\sum_{s=1}^{p}\left(\frac{(p+q-s)(p+q-s-1)!}{(p-s)!(q-1)!}\right) + \sum_{s=1}^{q}\left(\frac{(p+q-s)(p+q-s-1)!}{(q-s)!(p-1)!}\right)}{\sum_{s=1}^{p}\,{}^{p+q-(s+1)}C_{p-s} + \sum_{s=1}^{q}\,{}^{p+q-(s+1)}C_{q-s}}$$

Average running time -analysis

$$= \frac{\sum_{s=1}^{p} \left(\frac{q(p+q-s)!}{(p-s)!\ q!} \right) + \sum_{s=1}^{q} \left(\frac{p(p+q-s)!}{(q-s)!\ p!} \right)}{\sum_{s=1}^{p}\ {}^{p+q-(s+1)}C_{p-s} + \sum_{s=1}^{q}\ {}^{p+q-(s+1)}C_{q-s}}$$

Substitute $k = p-s$, $r = q-1$, $l = q-s$, $m = p-1$

$$= \frac{\sum_{k=0}^{p-1} q\ {}^{k+r+1}C_k + \sum_{l=0}^{q-1} p\ {}^{l+m+1}C_l}{\sum_{k=0}^{p-1}\ {}^{k+r}C_k + \sum_{l=0}^{q-1}\ {}^{l+m}C_l}$$

Average running time -analysis

Using the identity $\sum_{0 \le k \le n} {}^{r+k}C_k = \binom{r+n+1}{n}$

$$= \frac{q\binom{r+p+1}{p-1} + p\binom{m+q+1}{q-1}}{\binom{r+p}{p-1} + \binom{m+q}{q-1}}$$

Simplifying this gives $\dfrac{(p+q)\left[\frac{1}{q+1} + \frac{1}{p+1}\right]}{\left[\frac{1}{q} + \frac{1}{p}\right]} = \dfrac{p^2q + pq^2 + 2pq}{(p+1)(q+1)}$

Merge sort algorithm average running time -analysis

$$E(C) = \frac{p^2q + pq^2 + 2pq}{(p+1)(q+1)} = \frac{pq}{q+1} + \frac{pq}{p+1}$$

E(c) is the expected number of comparisons in the merge operation

p and q are length of sublists

Assume p = q = $\frac{n}{2}$

Recurrence relation : $T(n) = 2T\left(\frac{n}{2}\right) + \left[\frac{\frac{n^2}{4}}{1+\frac{n}{2}} + \frac{\frac{n^2}{4}}{1+\frac{n}{2}}\right]$

Merge sort algorithm average running time -analysis

$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + \left[\frac{\frac{n^2}{4}}{2+\frac{n}{2}}\right]$$

$$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + \left[\frac{\frac{n^2}{16}}{2+\frac{n}{4}}\right]$$

$$T(n) = 2\left[2T\left(\frac{n}{4}\right) + \frac{\frac{n^2}{4}}{2+\frac{n}{2}}\right] + \frac{n^2}{2+n}$$

$$T(n) = 4\left[2T\left(\frac{n}{8}\right) + \frac{\frac{n^2}{4}}{8+n}\right] + \frac{n^2}{4+n} + \frac{n^2}{2+n}$$

$$= 2^3T\left(\frac{n}{2^3}\right)+\frac{n^2}{2^3+n}+\frac{n^2}{2^2+n}+\frac{n^2}{2^1+n}$$

Merge sort algorithm average running time -analysis

Let $2^k = n$

$$= nT\left(\frac{n}{n}\right) + \sum_{k=1}^{lg\ n} \frac{n^2}{2^k + n}$$

$$= nT\left(1\right) + \sum_{k=1}^{lg\ n} \frac{n^2}{2^k + n}$$

$$=> \sum_{k=1}^{lg\ n} \frac{n^2}{2^k + n} \leq \int_{\mathbf{1}}^{lg\ n\ \mathbf{+1}} \frac{n^2}{2^x + n} dx$$

Merge sort algorithm average running time -analysis

$$\int_0^{lg\ n} \frac{n^2}{2^x + n} dx = n \left[x - \frac{log(n+2^x)}{log\ 2} \right]_0^{lg\ n} + C$$

$$= n \left[lg\ n - \frac{log(n+2^{lg\ n})}{log\ 2} - 0 + \frac{log(n+2^0)}{log\ 2} \right]$$

$$= n \left[lg\ n - \frac{log\ n + lg\ n\ log\ 2}{log\ 2} - 0 + \frac{log(n+1)}{log\ 2} \right]$$

$$\sum_{k=1}^{ln\ n} \frac{n^2}{2^k + n} \leq n\ [lg\ n]$$

$$=> T(n) \leq n + n\ lg\ n$$

$$T(n) = O(n\ lg\ n)$$

References

- Handbook of Theoretical Computer Science (North-Holland 1990). Chapter on Average Case Analysis of Algorithms and Data Structures : https://pdfs.semanticscholar.org/5fb8/cb9eb21663e6e1f2766a8fd094eb4758a743.pdf (section 3.8)

- Any book on Calculus