CSC 226
Zach Serviss
V00950002

# Assignment 2

1. A. Sample space = s{HHHH,THHH,HHHT,THHT,HHTH,THTH,HHTT,THTT,HTHH,TTHH,HTHT,TTHT,HTTH,TTTH,HTTT,TTTT}

B. Pr(A) = 1/2 (50%)
C. Pr(B) = 6/16 = 3/8 (37.5%)
D. Pr(A and B) = P(A)+P(B)-P(AB) = 1/2 + 3/8 - 3/16 = 11/16(68.75%)
E. E[X] = P(X=0) = 1/16, P(X=1) = 4/16, P(X=2) = 6/16, P(X=3) = 4/16, P(X=4) = 1/16
0(1/16)+1(4/16)+2(6/16)+3(4/16)+4(1/16) = 2
We would expect 2 heads to appear in 4 fair coin flips.

2. A. P = 1/4, 1/p = 1/(1/4) = 4. 4 flips until heads appears.
B. P(H==T) = (n choose n/2) (1/4)^n (3/4)^n-n/2 ; n is the number of coin flips

3.Given that we will go through our array of unsorted a maximum of n/2 times before we get a good pivot and the run time for partitioning is O(n). This gives us a runtime of O(n^2) For the recursive part of the equation we will take T(n/4) and T(3n/4), this is assuming we pick the worst good pivot and have a split of the array between 1/4 and 3/4. This gives the following recurrence equation of:

T(n) = T(n/4) + T(3n/4) + O(n^2)
T(n) = 2T(3n/4) + O(n^2)

a= 2, b = 4/3, c= 2

$\log_{4/3} 2 > 2$

Using the masters theorem this solves to $\Theta(n^{\log_{4/3} 2})$ which is the worst case time complexity for the given algorithm.

4.a

| Key | Value |
|---|---|
| 0 | |
| 1 | 20 |
| 2 | 16,5 |
| 3 | 44,88,11 |
| 4 | 94,39 |
| 5 | 12,23 |
| 6 | |

| Key | Value |
| --- | --- |
| 7 | |
| 8 | |
| 9 | 13 |
| 10 | |

b.

| Key | Value |
| --- | --- |
| 0 | 11 |
| 1 | 39 |
| 2 | 20 |
| 3 | 5 |
| 4 | 16 |
| 5 | 44 |
| 6 | 88 |
| 7 | 12 |
| 8 | 23 |
| 9 | 13 |
| 10 | 94 |

c.

| Key | Value |
| --- | --- |
| 0 | 16 |
| 1 | 11 |
| 2 | 88 |
| 3 | 20 |
| 4 | 23 |
| 5 | 44 |
| 6 | 94 |
| 7 | 12 |
| 8 | 39 |

| Key | Value |
|---|---|
| 9 | 13 |
| 10 | 5 |

d.

| Key | Value |
|---|---|
| 0 | 11 |
| 1 | 23 |
| 2 | 20 |
| 3 | 16 |
| 4 | 39 |
| 5 | 44 |
| 6 | 94 |
| 7 | 12 |
| 8 | 88 |
| 9 | 13 |
| 10 | 5 |

5. a = n/m (n keys, m slots) =load factor

Successful Search: $\Theta(1 + a)$
-this is the same as an unsorted linked list
Unsuccessful search: $\Theta(1 + a)$
-this will be slightly faster with a sorted list, but still have to factor in the load factor
Insertion: $\Theta(1+ a)$
-same as a successful search
Deletion: $\Theta(1)$ (if doubly linked list) or $\Theta(1+ a)$ (if singly linked list)

Professor Marley's hypothesis is incorrect, there is no substantial performance gains.