Name:_____

Date: _____

CSC 17C
Midterm

Problem 1) Analyze and compare the linear and binary search
- Show O( ) by mathematical analysis
- Show O( ) by timing comparison and graph results
- Show O( ) by operational analysis

Problem 2) Compare and contrast bubble sort with selection sort
- Show O( ) by mathematical analysis
- Show O( ) by timing comparison and graph results
- Show O( ) by operational analysis

Problem 3) Compare insertions i.e. push method with Simple Vector using arrays, with Optimized Simple Vector using arrays, and Simple Vector with Linked list.
- Show O( ) by mathematical analysis
- Show O( ) by timing comparison and graph results
- Show O( ) by operational analysis

Problem 4) Given $X = B^Y = H^Z$

$$\text{Prove} \quad \text{Log}_B(x) = \frac{\text{Log}_H(x)}{\text{Log}_H(B)}$$

Problem 5) Prove the summation formula

$$\sum_{i=1}^{N} i = 1+2+3 \cdots + (N-1) + N = \frac{N \cdot (N+1)}{2}$$

Problem 6) Derive the order of the error with respect to the exponential approximation.

$$e^{-1/N} \approx (1 - 1/N) \quad \text{where}$$

$$e^x = \sum_{i=0}^{\infty} x^i / i!$$

Problem 7) Derive the O() for the Recursive vs. non-Recursive Fibonacci function. Simple logic is enough using the fact that the Recursive Fibonacci takes Fibonacci time.

Name:_____ Date:_____

Problem 8) Given 4 dice calculate the probability of 1 pair, 2 pair, 3 of a kind and 4 of a kind. Simulate the results and compare to calculation.

Problem 9) Given a biased coin analogy, if a bit vector is 30% full, what are the odds that 5 bits randomly chosen all fall within the filled section. Simulate the results and compare to calculation.

Problem 10) Recursive function. Provide code and test sufficiently.

Derive a power function that is $O(\log_2 N)$

where $Z = X^N, \quad X \in R, \quad N \in N \rightarrow Z^+$

Hint: Split N for odd or even conditions

Problem 11) Recursive function. Provide code and test sufficiently.

Let $g(2x) = \dfrac{2g(x)}{1+g^2(x)}, \quad -1 \le x \le 1$

Base Condition

for $|x| < \varepsilon, \quad \varepsilon \approx 10^{-6}$ where $g(x) = x - x^3/6$

Problem 12) Mutual Recursion. Provide code and test sufficiently. $\approx x = \pi/4$

Let $C(2x) = 1/2\ C(x)\ S(x)$ and

$$S(2x) = \frac{C^2(x)\ S^2(x)}{C^2(x) - S^2(x)}$$

Base Condition

for $|x| < \varepsilon, \quad \varepsilon \approx 10^{-6}$

$S(x) = 1 + x^2/2$

$C(x) = 1/x + x/6$

Problem 13) Code the mode problem to utilize the Set and Map containers in the STL to solve for the number of modes in an array. Test and show correct solutions for no modes, 1 mode and multiple modes. Code with the Set to reduce the line count and number of for-loops necessary. Then utilize the Map to reduce the line count further.

Problem 1)

1. Binary Search

① W_n : the number of iterations of while loop in a worst-case execution

② If $2^i \le n < 2^{i+1}$, then $W_n = i+1$

$\rightarrow i = \log_2 n$, $W_n = \log_2 n + 1$

③ $O(\log_2 n)$

2. Linear Search

$\sum_{i=0}^{n-1} PO_3 = n \cdot PO_3 \Rightarrow O(n)$

(PO_3: if statement)

Time Comparison

Type 1 Linear Search
Type 2 Binary Search

(Unit: micro second)

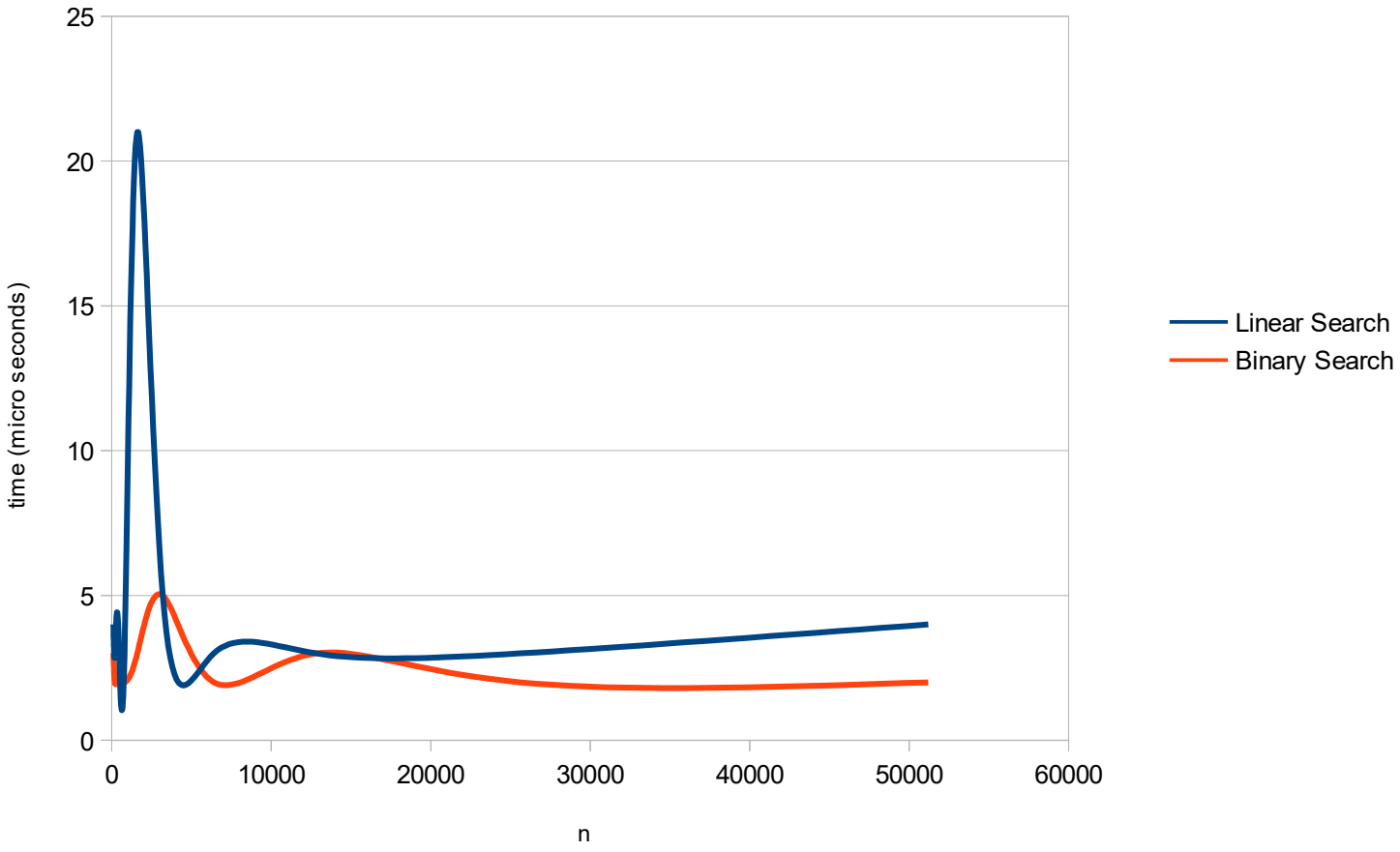
n	Type 1	Type 2
100	4	3
200	3	2
400	4	2
800	3	2
1600	21	3
3200	5	5
6400	3	2
12800	3	3
25600	3	2
51200	4	2

Operation Comparison

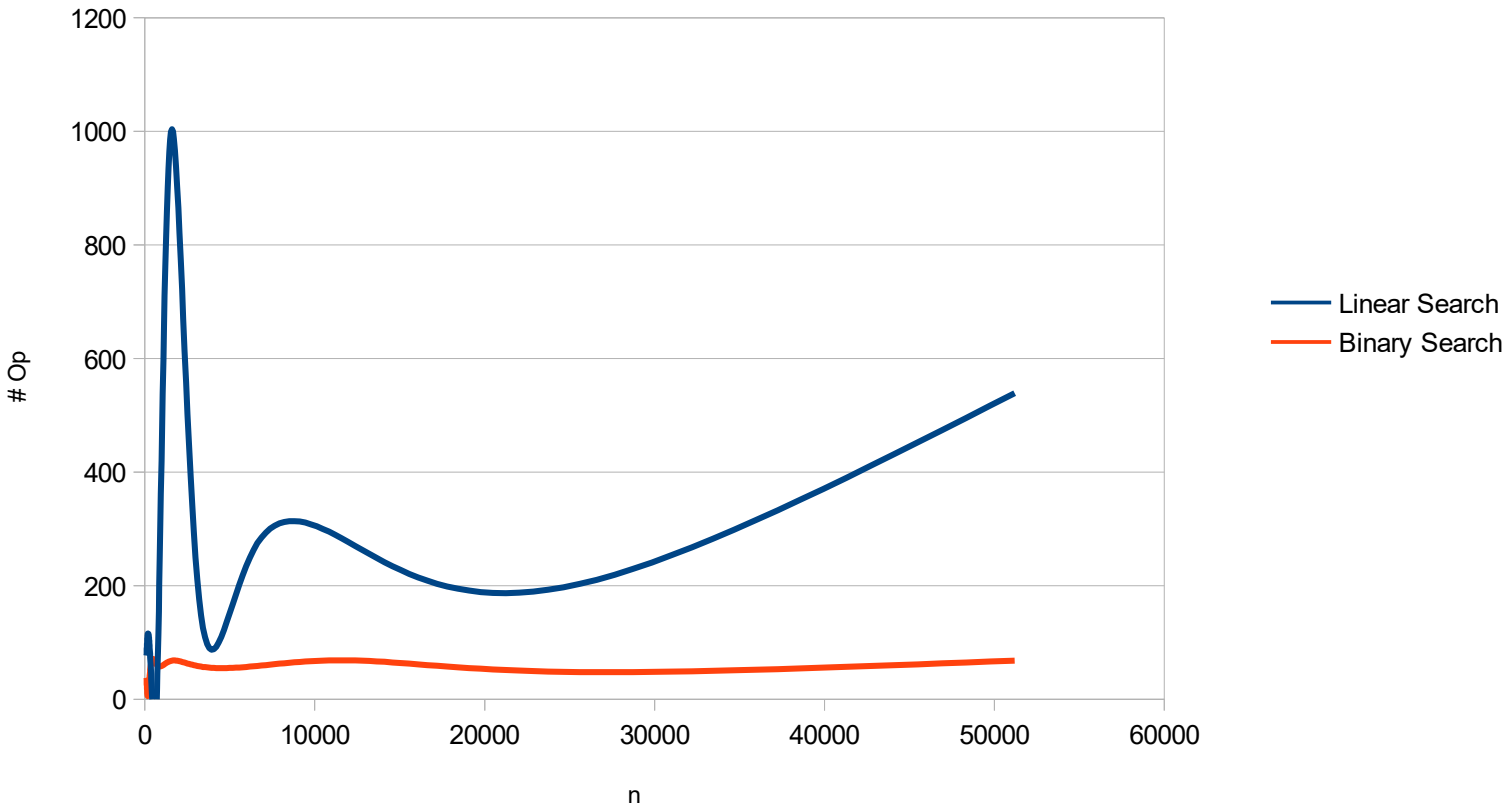
Type 1	Linear Search
Type 2	Binary Search

n	Type 1	Type 2
100	77	38
200	116	8
400	26	68
800	113	58
1600	1,004	68
3200	176	58
6400	263	58
12800	263	68
25600	203	48
51200	539	68

Time Comparison



Operation Comparison



Problem2)

1. Selection Sort

$$O_b + \sum_{i=0}^{n-1} (O_i + PO_s) + O_a$$

O_b: Before the loop; O_i: inner loop; PO_s: If statement; O_a: after the loop

$$= O_b + \frac{n(n-1)}{2} O' + O_a$$

$$= \frac{n^2}{2} O' - O' \frac{n}{2} + O_a + O_b$$

$$\Rightarrow \boxed{O(n^2)}$$

2. Bubble Sort

① W_n: the number of iterations of dowhile loop in a worst-case execution

② for loop. $\sum_{k=0}^{n-1} PO_s = (n) PO_s$

③ $\Rightarrow (n-2)\, n \cdot PO_s$

$\therefore O(n^2)$

Time Comparison

Type 1 Bubble Sort
Type 2 Selection Sort

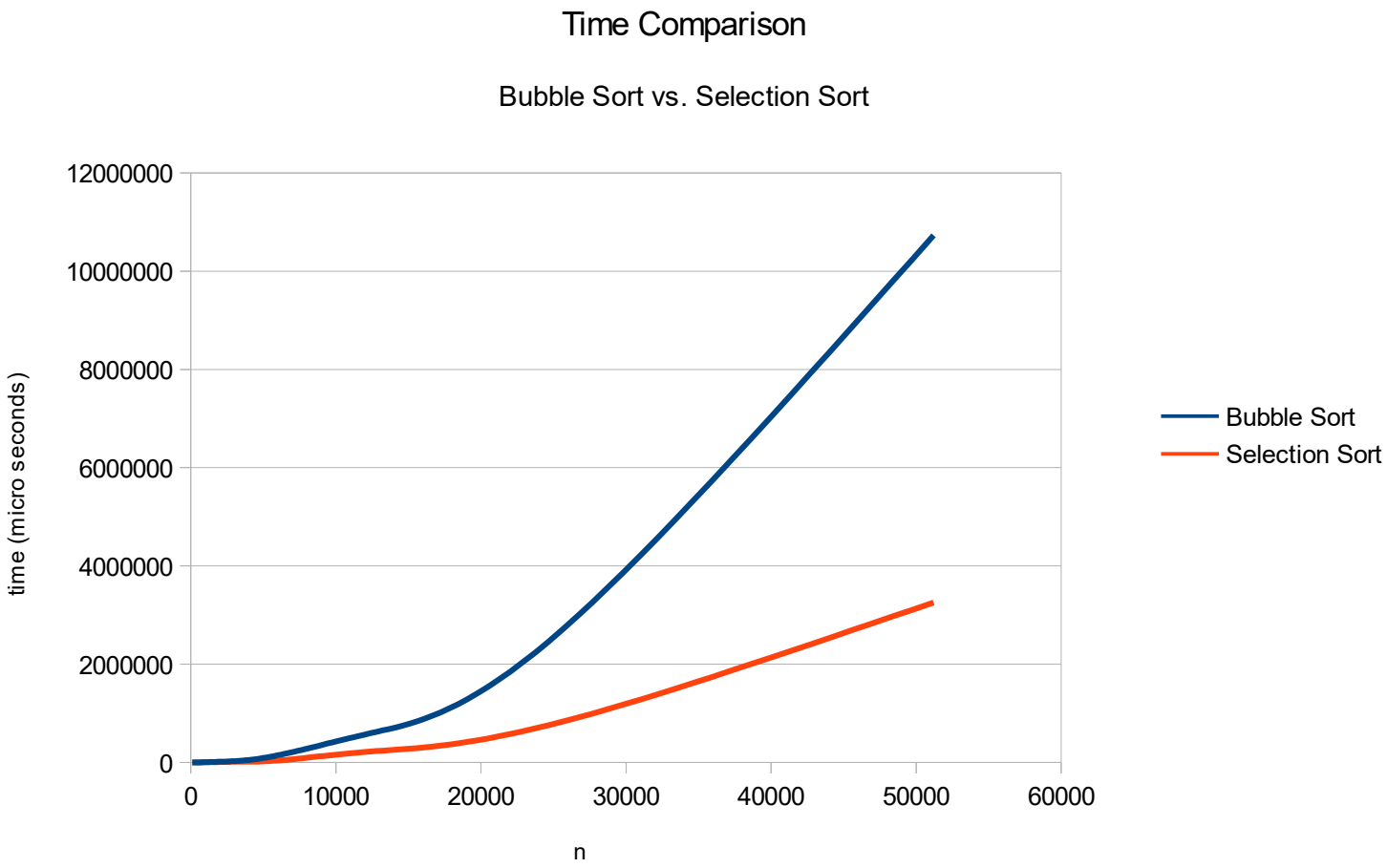
(Unit: micro second)

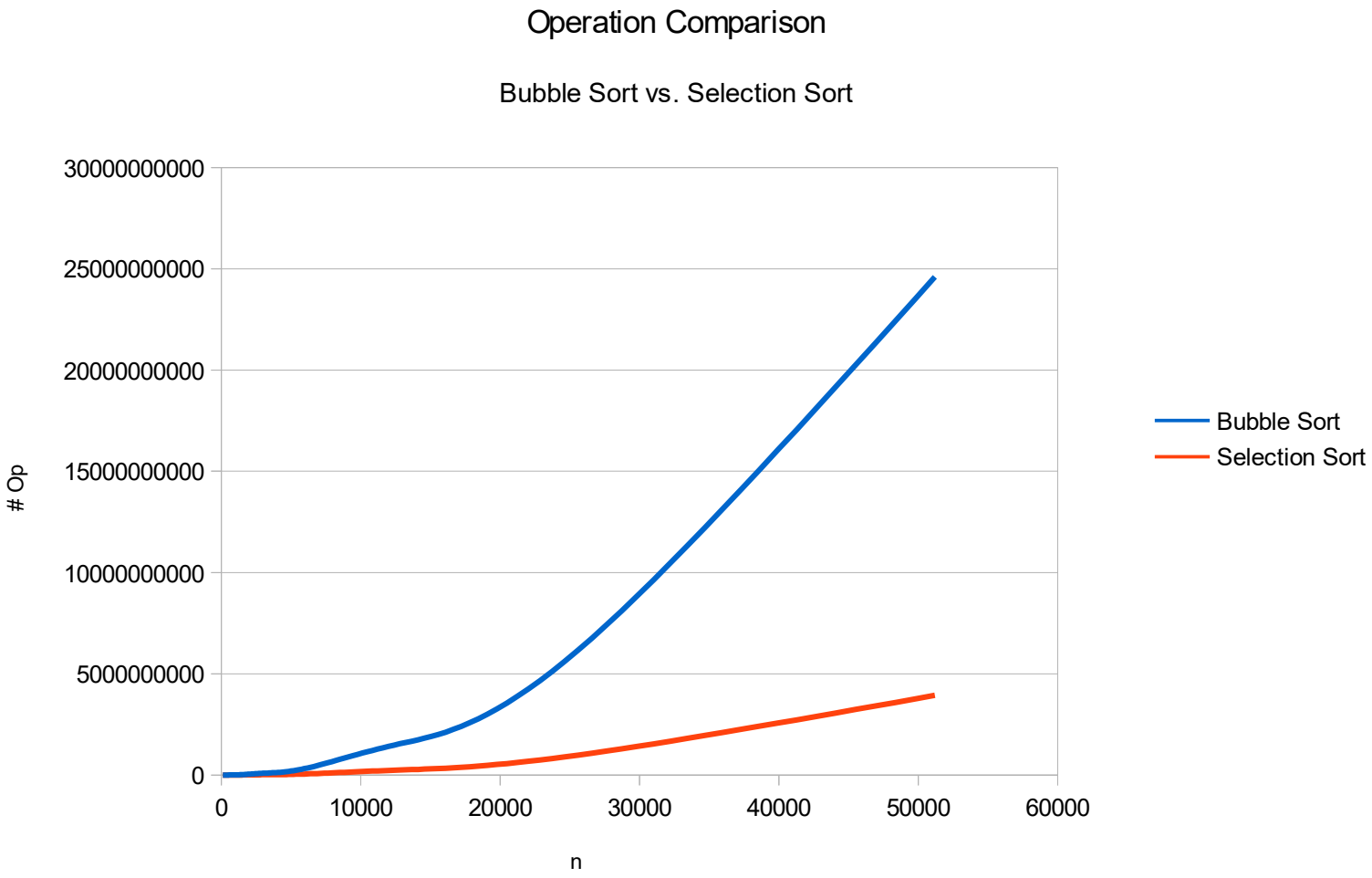
n	Type 1	Type 2
100	46	19
200	180	86
400	706	290
800	2,244	1,218
1600	10,391	4,814
3200	29,876	12,512
6400	169,078	47,740
12800	621,179	229,346
25600	2,697,879	827,399
51200	10,730,047	3,255,863

Operation Comparison

Type 1	Bubble Sort
Type 2	Selection Sort

n	Type 1	Type 2
100	80,248	16,933
200	366,304	64,030
400	1,495,626	248,785
800	5,874,208	977,713
1600	23,489,446	3,876,397
3200	94,893,812	15,433,258
6400	383,158,458	61,585,522
12800	1,541,414,186	246,057,775
25600	6,167,223,228	983,628,193
51200	24,598,212,394	3,933,343,522


Time Comparison
Bubble Sort vs. Selection Sort
12000000
10000000
8000000
6000000
4000000
2000000
0
time (micro seconds)
0
10000
20000
30000
40000
50000
60000
n
Bubble Sort
Selection Sort


Operation Comparison
Bubble Sort vs. Selection Sort
30000000000
25000000000
20000000000
15000000000
10000000000
5000000000
0
Op
0
10000
20000
30000
40000
50000
60000
n
Bubble Sort
Selection Sort

Problem3)

1. SimpleVector.

$O_b + \sum_{i=0}^{n-1} O_i + O_a$

O_b: before the loop; $\sum_{i=0}^{n-1} O_i$: for loop; O_a: after the loop

$= \frac{n(n-1)}{2} + O_a + O_b.$

$\Rightarrow O(n^2)$

2. Simple Vector Push Efficient

① if (maxSize == arraySize)

$\sum_{i=0}^{arraySize} O_i$

② else $O(1)$

In the beginning, since the array size is relatively small it doesn't affect to the efficiency and as the array size becomes larger, the case to create additional memory space, which uses for-loop, rarely happens. In this reason, its graph is y=c where c is constant in the simulation.

3. Simple Vector Using linked list

① addList

$W_n = n$ (worst-case of while loop)

$\Rightarrow \boxed{O(n)}$

Time Comparison

Type 1	Simple Vector using arrays
Type 2	Optimized Simple Vector using arrays
Type 3	Simple Vector with Linked list

(Unit: micro second)

n in for-loop in main function	n	Type 1	Type 2	Type 3
	464	14,690		
	446		434	
100	439			145
	915	53,588		
	898		743	
200	906			310
	1347	114,943		
	1346		1,347	
300	1442			812
	1754	190,953		
	1845		1,400	
400	1760			912
	2277	280,202		
	2235		2,285	
500	2269			988
	2856	424,997		
	2628		3,179	
600	2752			1,967
	3059	538,281		
	3109		3,521	
700	3078			2,101
	3521	793,849		
	3695		2,862	
800	3528			3,037
	4083	903,452		
	4006		3,865	
900	3972			2,996
	4393	1,120,424		
	4594		3,183	
1000	4398			4,640
	4993	1,365,880		
	4848		5,740	
1100	4930			4,092
	5387	1,846,003		
	5499		5,426	
1200	5520			6,543
	5731	1,918,461		
	5981		6,165	

Time

Type 1	Simple Vector using arrays
Type 2	Optimized Simple Vector using arrays
Type 3	Simple Vector with Linked list

(Unit: micro second)

n in for-loop in main function	n	Type 1	Type 2	Type 3
1300	5829			6,525
	6238	2,275,923		
	6363		6,551	
1400	6264			7,113
	6812	2,583,400		
	6867		6,837	
1500	6552			9,580
	7109	2,904,343		
	7232		7,582	
1600	7281			8,978
	7513	3,286,578		
	7940		8,407	
1700	7515			11,209
	8072	3,609,546		
	8306		7,332	
1800	7944			12,441
	8460	3,937,719		
	8491		6,668	
1900	8691			11,277
	9167	4,405,148		
	9091		6,278	
2000	8949			16,604
5000	22335	27,253,067		
	45105		52,451	
10000	44556			534,780
	225852		188,903	
100000	224950			12,650,044

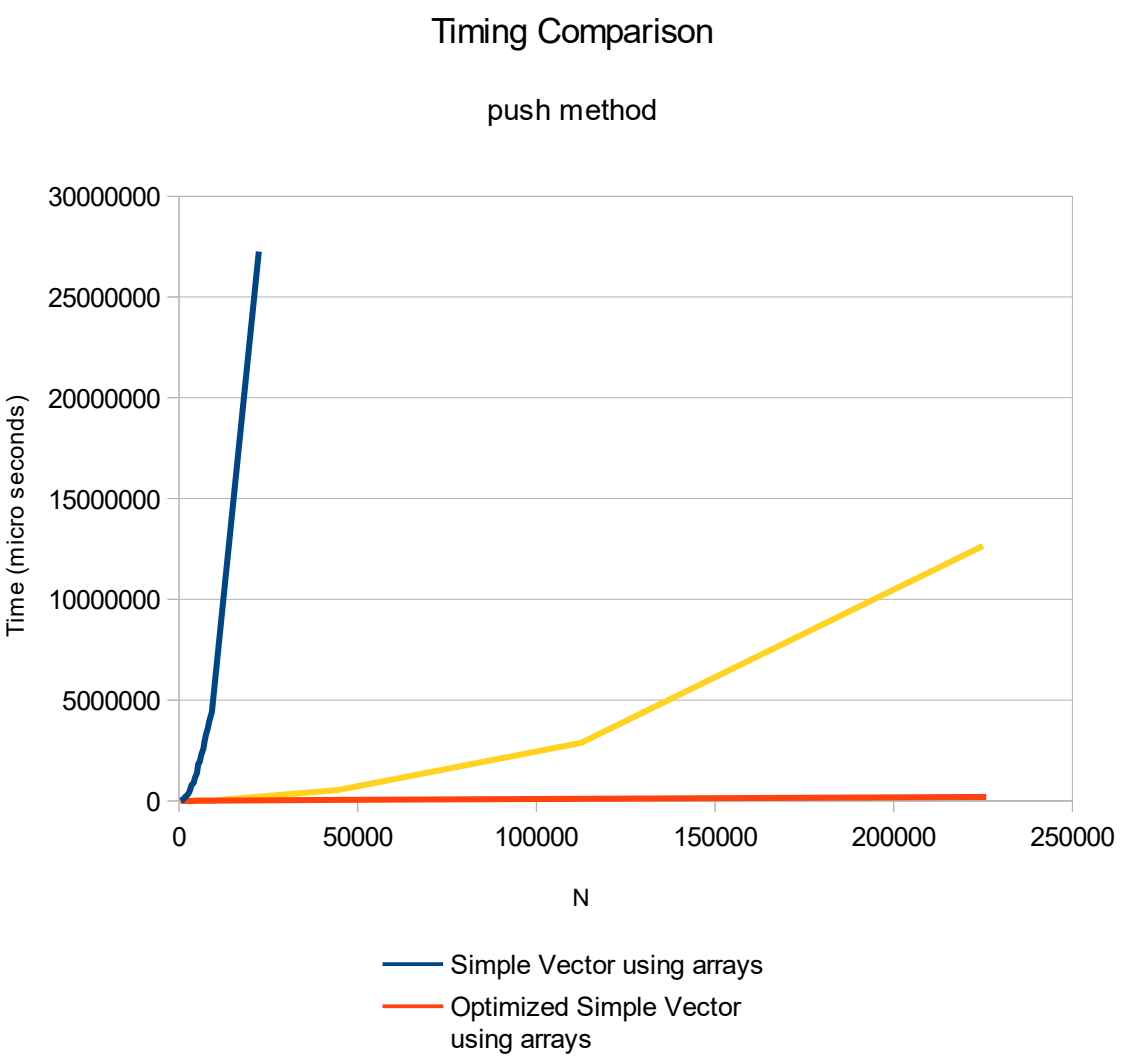
Operation Comparison

Type 1	Simple Vector using arrays
Type 2	Optimized Simple Vector using arrays
Type 3	Simple Vector with Linked list

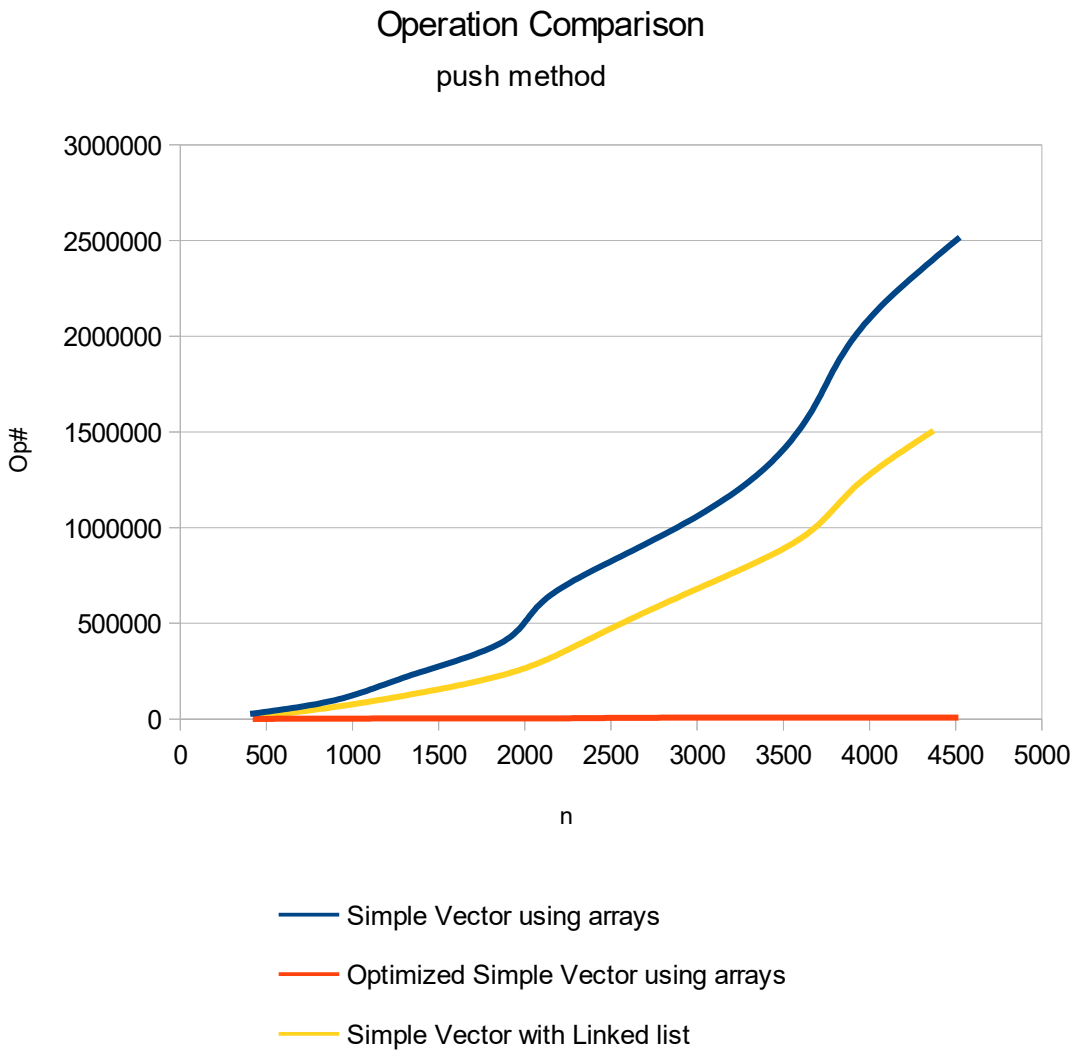
(Unit: micro second)

n in for-loop in main function	n	Type 1	Type 2	Type 3
	405	26,664		
	421		994	
100	467			15,650
	919	103,314		
	885		1,942	
200	883			61,300
	1342	229,964		
	1364		3,530	
300	1391			136,950
	1878	406,614		
	1855		3,830	
400	1920			242,600
	2129	633,264		
	2237		4,130	
500	2286			378,250
	2691	909,914		
	2740		6,998	
600	2665			543,900
	3295	1,236,564		
	3226		7,298	
700	3149			739,550
	3667	1,613,214		
	3605		7,598	
800	3635			965,200
	3949	2,039,864		
	4086		7,898	
900	3925			1,220,850
	4523	2,516,514		
	4515		8,198	
1000	4371			1,506,500

plot_Time


Timing Comparison
push method
30000000
25000000
20000000
15000000
10000000
5000000
0
Time (micro seconds)
0
50000
100000
150000
200000
250000
N
Simple Vector using arrays
Optimized Simple Vector using arrays

plot_Op


Operation Comparison
push method
3000000
2500000
2000000
1500000
1000000
500000
0
Op#
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
n
Simple Vector using arrays
Optimized Simple Vector using arrays
Simple Vector with Linked list

Problem4)

① Left Hand Side or LHS of Equation

② $\log_B X = \frac{\log X}{\log B}$

② Right Hand Side or RHS of Equation

$$\frac{\log_H X}{\log_H B} = \frac{\frac{\log X}{\log H}}{\frac{\log B}{\log H}}$$

$$= \frac{\log X}{\log B}$$

③ $LHS \stackrel{?}{=} RHS$

$$\frac{\log X}{\log B} = \frac{\log X}{\log B} \quad \checkmark$$

Therefore, $\log_B X = \frac{\log_H X}{\log_H B}$

Problem5)

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n-1 + n$$

$$+ \quad \sum_{i=1}^{n} i = n + n-1 + n-2 + \cdots + 2 + 1 \quad \text{(Reverse order)}$$

$$2\sum_{i=1}^{n} i = \underbrace{(1+n) + (1+n) + (n+1) + \cdots + (n+1) + (n+1)}_{n}$$

$$= n(n+1)$$

Therefore, $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Problem6)

$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

$$= \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \cdots$$

$$= 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \cdots$$

So, $e^{-\frac{1}{n}} = 1 - \frac{1}{n} + \frac{\left(-\frac{1}{n}\right)^2}{2} + \frac{\left(-\frac{1}{n}\right)^3}{6} + \cdots$

$$= 1 - \frac{1}{n} + \frac{1}{2n^2} - \frac{1}{6n^3} + \cdots$$

$$= 1 - \frac{1}{n}\Big(\underbrace{1 - \frac{1}{2n}\Big(1 - \frac{1}{3n}\Big(1 - \frac{1}{4n}\Big(1 - \frac{1}{5n}\cdots}_{\approx 1}$$

$$\approx 1$$

$$\approx 1 - \frac{1}{n}$$

Problem7)

In the simulation,

The number of calling fiboRec(0) + The number of calling fiboRec(1)

= fiboRec(n+1)

So, it takes Fibonacci time when we execute the recursive Fibonacci.

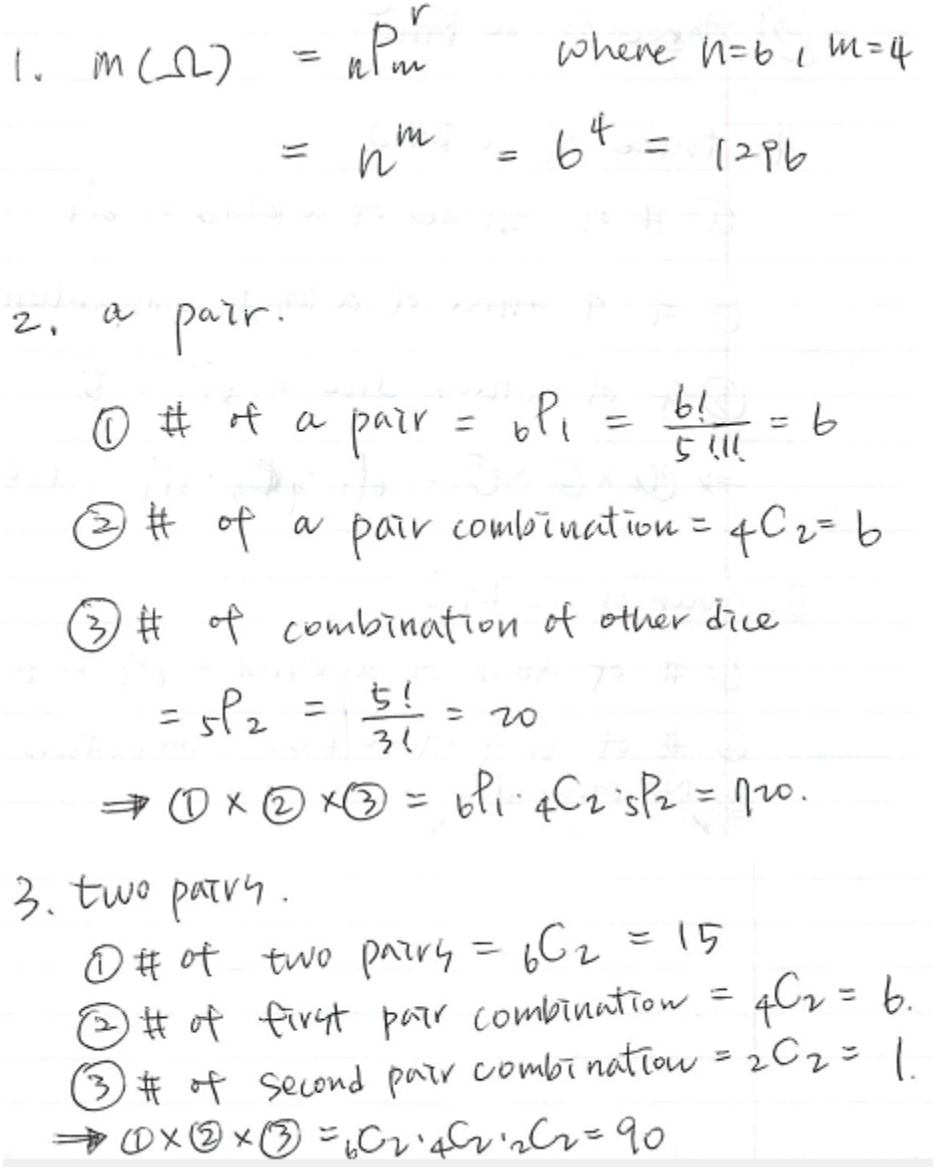
Fibonacci Recursion

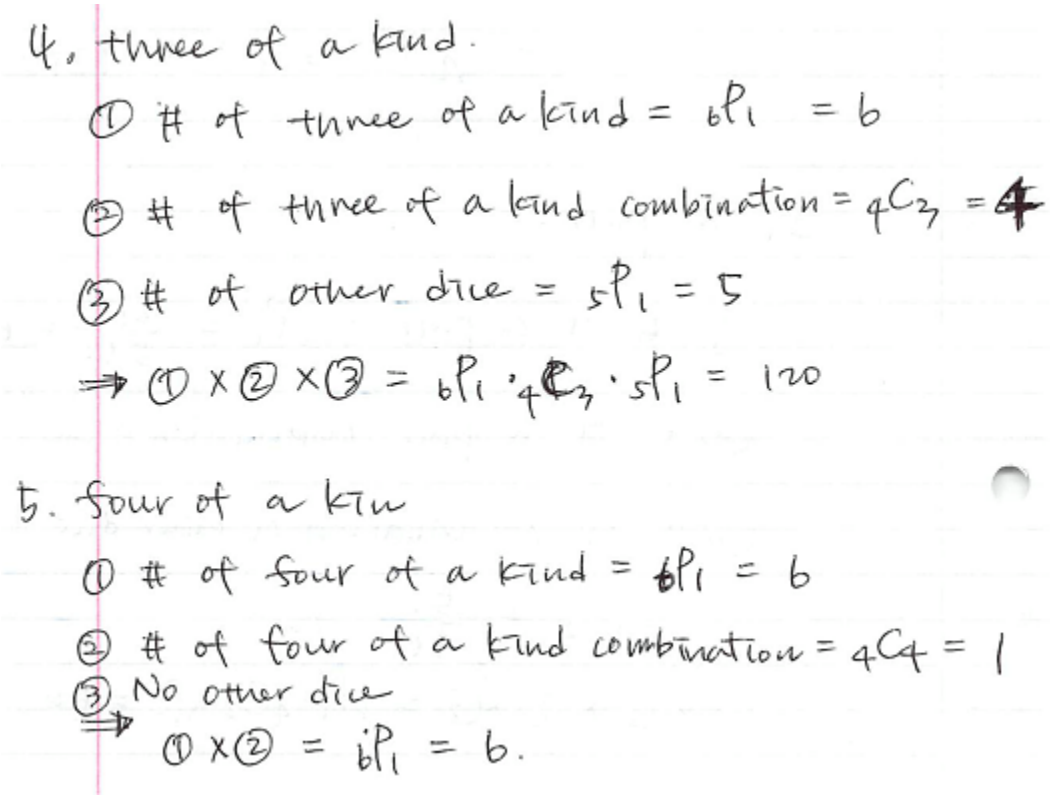
n	fiboRec(n)	#0 Call func(0) if(n==0)	#1 Call func(1) Else if(n==1)	#0 + #1 Total # of Call
0	0	1	0	1
1	1	0	1	1
2	1	1	1	2
3	2	1	2	3
4	3	2	3	5
5	5	3	5	8
6	8	5	8	13
7	13	8	13	21
8	21	13	21	34
9	34	21	34	55
10	55	34	55	89
11	89	55	89	144
12	144	89	144	233
13	233	144	233	377
14	377	233	377	610
15	610	377	610	987
16	987	610	987	1597
17	1597	987	1597	2584
18	2584	1597	2584	4181
19	4181	2584	4181	6765
20	6765	4181	6765	10946
21	10946	6765	10946	17711
22	17711	10946	17711	28657
23	28657	17711	28657	46368
24	46368	28657	46368	75025
25	75025	46368	75025	121393
26	121393	75025	121393	196418
27	196418	121393	196418	317811
28	317811	196418	317811	514229
29	514229	317811	514229	832040
30	832040	514229	832040	1346269
31	1.34627e+06	832040	1346269	2178309
32	2.17831e+06	1346269	2178309	3524578
33	3.52458e+06	2178309	3524578	5702887
34	5.70289e+06	3524578	5702887	9227465
35	9.22746e+06	5702887	9227465	14930352
36	1.49304e+07	9227465	14930352	24157817
37	2.41578e+07	14930352	24157817	39088169
38	3.90882e+07	24157817	39088169	63245986
39	6.3246e+07	39088169	63245986	102334155
40	1.02334e+08	63245986	102334155	165580141
41	1.6558e+08	102334155	165580141	267914296

42	2.67914e+08	165580141	267914296	433494437
43	4.33494e+08	267914296	433494437	701408733
44	7.01409e+08	433494437	701408733	1134903170
45	1.1349e+09	701408733	1134903170	1836311903
46	1.83631e+09	1134903170	1836311903	2971215073

Problem8)

1. Calculation:





2. Simulation:

Pair	number of event	%	Theoretical
a Pair	585	58.5	55.5556
two Pairs	67	6.7	6.94444
three of a kind	81	8.1	9.25926
four of a kind	7	0.7	0.462963

```
RUN SUCCESSFUL (total time: 1s)
```

3. Sumarry

	Theoretical		Simulation		
	#	Probability	#	Probability	Diff
a pair	720	56%	585	59%	-3%
two pair	90	7%	67	7%	0%
three of a kind	120	9%	81	8%	1%
four of a kind	6	0%	7	1%	0%

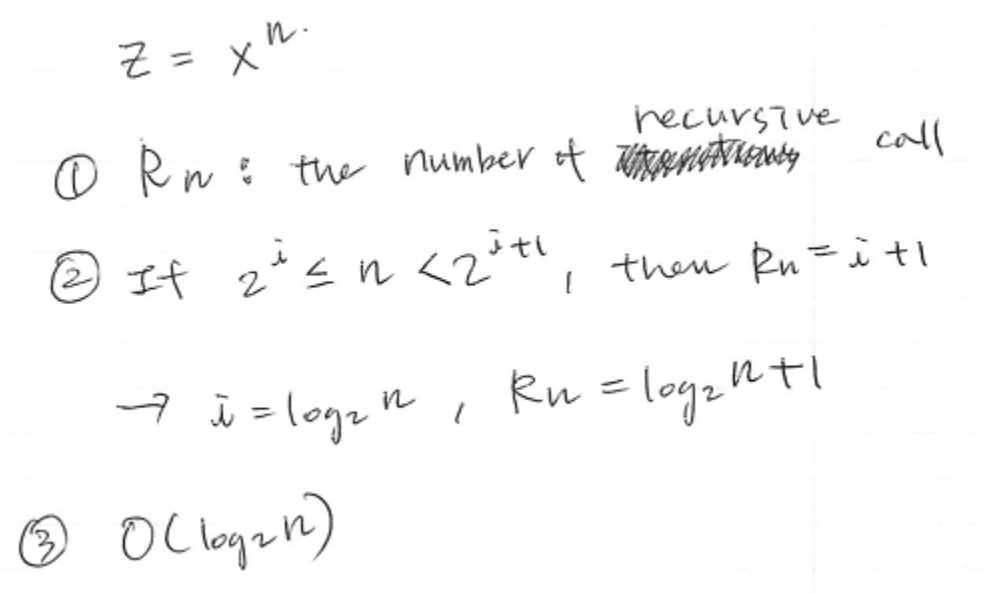
Problem9)

```
a) (1) p = Probability getting Head = 3/10, q = Probability getting tail = 7/10
   (2) (p+q)^n = (sigma sign) from i=0 to i=n (n i) p^i * q^(n-i) = 1
        (n i) is n choose i combination
   (3) So, if we substitute i=5, n=5, p=0.3, q=0.7
       then (0.3)^5 * (0.7)^0 = 0.00243, that is, 0.243%
b) Simulation result:0.243%

RUN SUCCESSFUL (total time: 700ms)
```

Problem10, 11, 12)

1. Calculation (Problem10):



2. Simulation Result(Problme10, 11, 12)

```
Problem10: (1) x=30.5, n=5 (2) x^n= 2.63936e+07
Problem11: (1) x=0.5 (2) g(x)= 0.244919
Problem12: (1) x=PI/4 (2) C(x)= 1.41421, S(x)= 1.41421

RUN SUCCESSFUL (total time: 5s)
```

Problem13)

When arySize=50 and modNum=12, simulation result:

```
0 1 2 3 4 5 6 7 8 9
10 11 0 1 2 3 4 5 6 7
8 9 10 11 0 1 2 3 4 5
6 7 8 9 10 11 0 1 2 3
4 5 6 7 8 9 10 11 0 1


Mode Freq = 5
Number of modes = 2

The number of modes = 2
The max Frequency = 5
The mode set = {0,1}

RUN SUCCESSFUL (total time: 103ms)
```