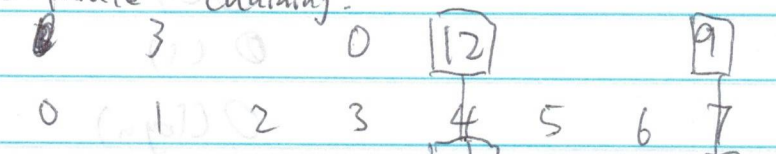
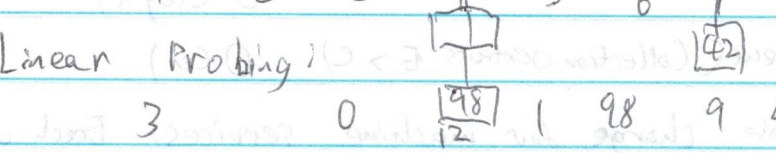
YI CHOU

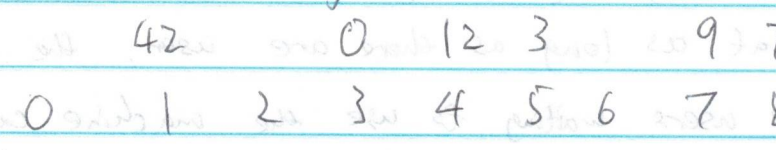
1. Separate chaining:



Linear Probing:



Quadratic Probing:



2. I will choose 500., Because I don't know the total entries in the hash table so maximum size is the best choice.

3. (1) 53491/106963 = 0.5 entries per bucket

(2) Yes, hashtable should rehash when the total number of entries is 25% or less of your total size.

(3) NO, if will be better to rehash ~~the~~ when loadfactor =0.75.

4.

Insert (x)	O(1)
Rehash ( )	O(N)
Remove (x)	O(1)
Contains (x)	O(1)

7. I think the problem is the ArrayList<HashItem<T>> oldArray =array; Because It it will ~~~~ make a deep copy of 'array' and that will be expensive when table cells get filled over 50% when you rehash the table.

8.

push (x)	O(1)
top()	O(1)
pop()	O(log n)
priorityQueue(Collection<extends E>c)	O(n)

9. Suppose people charge for machine services. Each user pay the same fee every time for using the machine, but different user need different service time. In order to obtain the maximum profit, suppose that as long as there are user, the machine will not be idle, the users waiting to use the machine can be organized in to a minimum priority queue, and the priority is the service time required by the user. When a new user needs to use the machine. Request will be added to the priority queue.

10. parent: $\frac{i}{2}$

left child: $2i$

right child: $2i+1$

11. 10

10 12

1 12 10

1 12 10 14

1 6 10 14 12

1 6 5 14 12 10

~~1 6 5 14 12 10 15~~

1 3 5 6 12 10 15 14

1 3 5 6 12 10 15 14 11

12. 1 3 5 6 12 10 15 14 11

13. 3 6 5 11 12 10 15 14

5 6 10 11 12 14 15

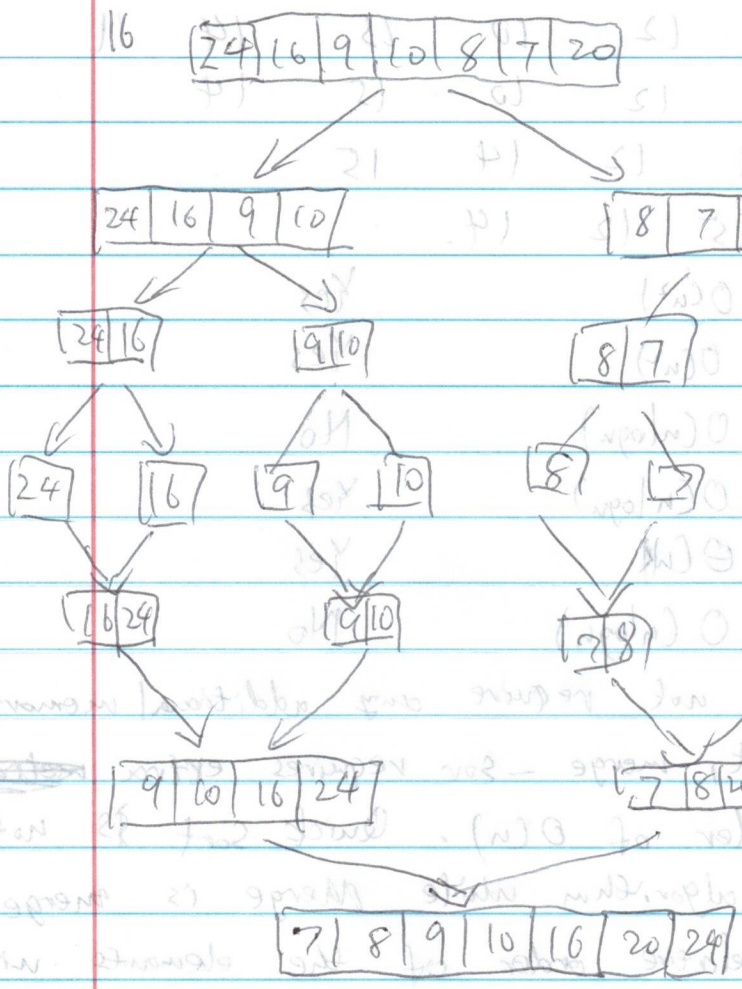
6 11 10 15 12 14.

14.

Bubble sort	$O(n^2)$	Yes
Insertion Sort	$O(n^2)$	Yes
Heap Sort	$O(n \log n)$	No
Merge Sort	$O(n \log n)$	Yes
Radix Sort	$\Theta(nk)$	Yes
Quick Sort	$O(n \log n)$	No

15. Quick sort does not require any additional memory while sorting but, merge - sor requires extra ~~memory~~ memory in order of $O(n)$. Quick sort is not a stable sorting algorithm while Merge is merge sort presorves the relative order of the elements with equal values but, quick sort does not guarantee that when compared with running times quick sort is a bit faster than merge sort and uses no extra memory. Thus influencing languages to dose quick sort over mergesort.

16



17. 24 16 9 8 7 20 10

7 16 9 8 24 20 10

7 8 9 16 24 20 10

7 8 9 10 24 20 16

7 8 9 10 24 16 20

7 8 9 10 16 24 20

7 8 9 10 16 20 24.