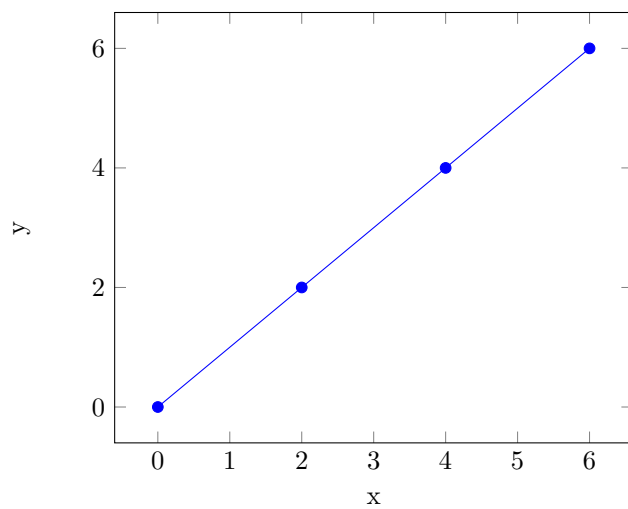
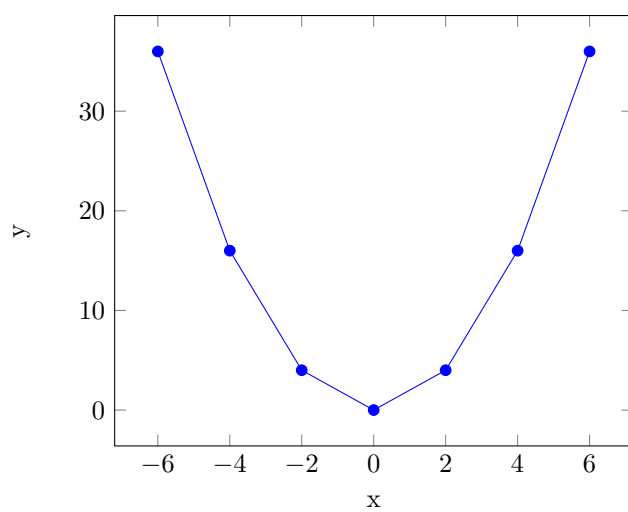


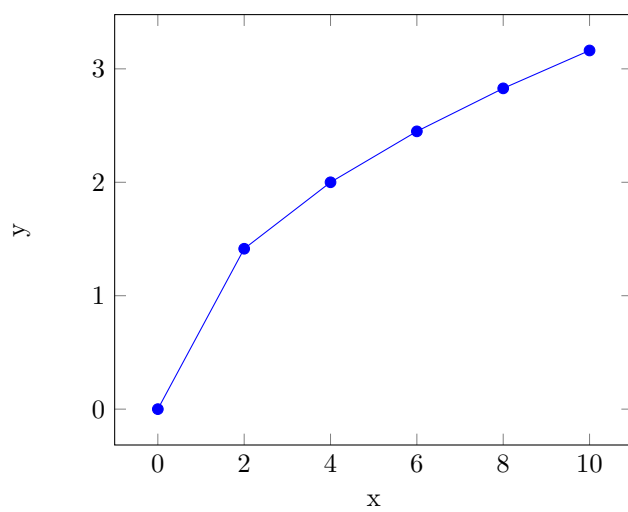
1.
 - \mathcal{O} - Pior caso, pois nele mostra o tempo máximo que o algoritmo irá completar a tarefa.
 - Ω - Melhor caso para execução de uma tarefa.
 - Θ - Complexidade média de execução.
2.
 - (a) 1
 - (b) 2
 - (c) 4
 - (d) 8
 - (e) 16
 - (f) 32
 - (g) 64
 - (h) 128
 - (i) 256
 - (j) 512
 - (k) 1024
 - (l) 2048
 - (a) n
 - (b) 10
 - (c) 9
 - (d) 8
 - (e) 7
 - (f) 6
 - (g) 5
 - (h) 4
 - (i) 3
 - (j) 2
 - (k) 1
 - (l) 0
 - (a) 5
 - (b) 4
 - (c) 5
 - (d) 4
 - (e) 4
 - (f) 4
 - (g) 4,...
 - (h) 5
 - (i) 4
 - (j) 3,...
 - (k) 4
 - (l) 3



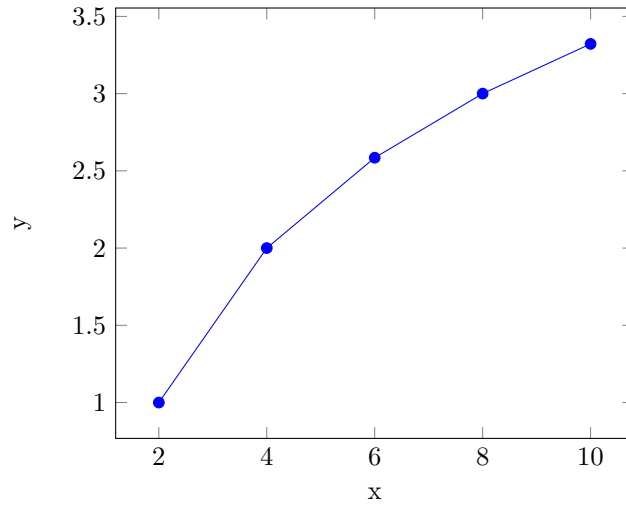
• (a)



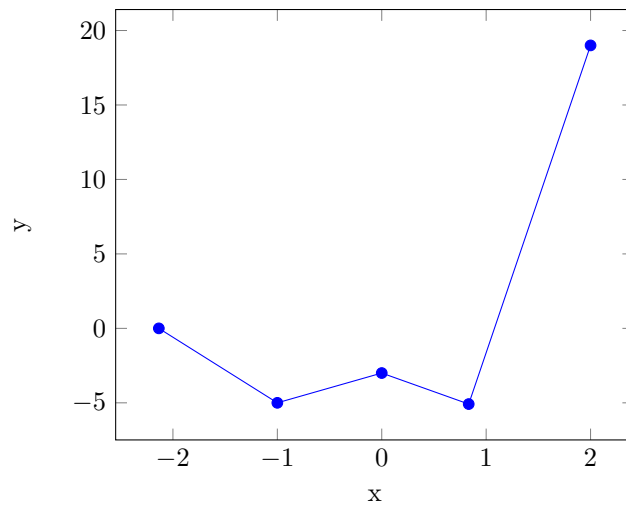
(b)



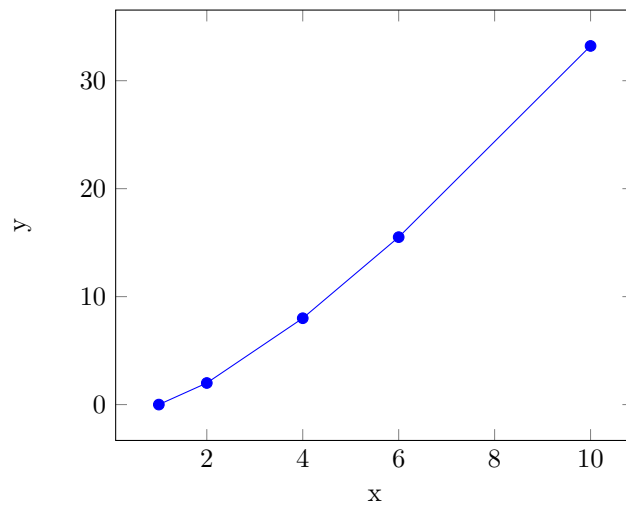
(c)



(d)



(e)



(f)

3. $3 = O(1)$

4. 5 ou $3 = O(1)$

5. MC- $5 = \Omega(1)$ — PC- $7O(1)$

6. $4 = O(1)$
7. $2n = O(n)$
8. $3 = O(1)$
9. $n-3 = O(n)$
10. $4 = O(1)$
11. $4 = O(1)$
12. $17 \text{ ou } 8 = O(1)$
13. $6 = O(1)$
14. $n^2 = O(n^2)$
15. $9 = O(1)$
16. $n^2 - 3n = O(n^2)$
17. $n^2 - 6n = O(n^2)$
18. $\log_2(n) = O(\log_2(n))$
19. $\log_2(n + 5) = O(\log_2(n))$
20. $n^2 - 14n + 49 = O(n^2)$
21. $\log_2(n + 2) = O(\log_2(n))$
22. $\log_2(n) = O(\log_2(n))$
23. $\log_2(n) = O(\log_2(n))$
24. $\log_2(n + 1) = O(\log_2(n))$

25. (a)

```

1      for ( i=0; i<n; i++){
2          a--;
3          b--;
4          c--;
5      }
```

(b)

```

1      for ( i=0; i<n; i++){
2          for (j=0; i<n; j++){
3              a--;
4              b--;
5              c--;
6          }
7      }
```

(c)

```

1      for ( i=0; i<n; i++){
2          a--;
3          b--;
4          c--;
5          d--;
6          e--;
7      }

```

(d)

```

1      for ( i=0; i<n; i++){
2          for (j=0; i<n; j++){
3              for ( f=0; f<n; f++){
4                  a--;
5                  b--;
6                  c--;
7              }
8          }
9      }

```

(e)

```

1      for ( i=1+n; i >0; i/=2){
2          a--;
3      }

```

(f)

```

1      for ( i=0; i<n; i++){
2          for (j=0; i<n; j++){
3              for ( f=0; f<n; f++){
4                  a--;
5                  b--;
6              }
7          }
8      }
9      for ( i=0; i<5; i++){
10         a--;
11     }

```

- 26.
- Comparação entre elementos do array.
 - $n - 1$ vezes.
 - Para o pior caso.