COSTI COMPUTAZIONALI C

|  |  |  |  |
| --- | --- | --- | --- |
| Malloc(size) | O(1) | Costante | Sempre |
| Free(elemento) | O(1) | Costante | Sempre |
| Realloc | O(n) | Lineare | Caso peggiore |

COSTI ARRAY DINAMICI

|  |  |  |  |
| --- | --- | --- | --- |
| Append | O(1) | Costante | Caso medio |
| Pop | O(1) | Costante | Caso medio |
| Append | O(n) | Lineare | Caso peggiore |
| Pop | O(n) | Lineare | Caso peggiore |
| Copia array | O(n) | Lineare | Sempre |
| Insert | O(n) | Lineare | Caso peggiore |
| Insert | O(n-p) | Lineare in n-p | Caso medio |

COSTI LINKED LIST

|  |  |  |  |
| --- | --- | --- | --- |
| In\_0 | O(1) | Costante | Sempre |
| In\_1 | O(1) | Costante | Sempre |
| List\_out\_0 | O(1) | Costante | Sempre |
| List\_out\_1 | O(1) | Costante | Sempre |
| N inserimenti con puntatore | O(n) | Lineare | Sempre |
| Append | O(n) | Lineare | Sempre |
| Insert | O(n) | Lineare | Caso peggiore |
| Search | O(n) | Lineare | Caso peggiore |
| N inserimenti con append | O(n^2) | Quadratico | sempre |

DIZIONARI CON LINKED LIST

|  |  |  |  |
| --- | --- | --- | --- |
| Dict\_search\_chiave\_k | O(1) | Costante | Caso medio |
| Inserimento | O(1) | Costante | Caso medio |
| Cancellazione 1 elemento | O(1) | Costante | Caso medio |
| Dict\_search\_chiave\_k | O(n) | Lineare | Caso peggiore |
| Inserimento | O(n) | Lineare | Caso peggiore |
| Inserimento con funzione hash | O(n) | Lineare | Caso medio |
| Cancellazione di 1 elemento | O(n) | Lineare | Caso peggiore |
| Creazione | O(n) | Lineare | Sempre |
| Print di tutti gli elementi | O(n) | Lineare | Sempre |

COSTI COMPUTAZIONALI PYTHON

COSTI LISTE (a= lista di grandezza O(n))

|  |  |  |  |
| --- | --- | --- | --- |
| a.append | O(1) | Costante | Caso medio |
| Len(a) | O(1) | Costante | Sempre |
| Indexing | O(1) | Costante | Sempre |
| x in a | O(n) | Lineare | Sempre |
| min/max | O(n) | Lineare | Sempre |
| Del(a[i]) | O(n) | Lineare | Caso peggiore |
| Slicing | O(m)  ->m=sottolista | Lineare nella sottolista | sempre |

COSTI SET (a,b= 2 set di grandezza O(n))

|  |  |  |  |
| --- | --- | --- | --- |
| Creazione set vuoto | O(1) | Costante | Sempre |
| x in a | O(1) | Costante | Sempre |
| a.add(x) | O(1) | Costante | Sempre |
| a.remove(x) | O(1) | Costante | Sempre |
| a|b->unione | O(n) | Lineare | Sempre |
| a&b->intersezione | O(n) | Lineare | Sempre |
| a-b->differenza | O(n) | Lineare | Sempre |

COSTI DICT

|  |  |  |  |
| --- | --- | --- | --- |
| Get(->print(d[x])) | O(1) | Costante | Caso medio |
| Del(->del(d[x])) | O(1) | Costante | Caso medio |
| Upgrade  (->d[x]=”nuovo valore ”) | O(1) | Costante | Caso medio |
| Inserimento(d[x]) | O(1) | Costante | Caso medio |
| Appartenenza  (->”x” in d) | O(1) | Costante | Caso medio |
| Appartenenza  (-> “x” in d) | O(n) | Lineare | Caso peggiore |
| Iterazioni(cicli) | O(n) | Lineare | Sempre |
| Metodi che restituiscono sequenze di: chiavi, valori e elementi | O(n) | Lineare | Sempre |

COSTI ALGORITMI:

|  |  |  |  |
| --- | --- | --- | --- |
| nome | caso migliore | caso peggiore | caso medio |
| bubble-sort | O(n) | O(n^2) | O(n) |
| merge-sort | O(nln(n)) | O(nln(n)) | O(nln(n)) |
| merge | O(n) | O(n) | O(n) |
| binary-search | O(1) | O(ln(n)) | O(ln(n)) |
| insertion-search | O(n) | O(n^2) | O(n^2) |

PRETEST:

quando siamo davanti a questa istruzione: int \*b=&a allora ogni modifica fatta sia su a che su b viene applicata anche all’altra, in quanto b diventa un puntatore all’indirizzo di memoria di a.  
SPAZI MEMORIA:  
char-> 11 byte, int-> 4 byte, float-> 4 byte, double-> 8 byte. Puntatore -> 8 byte(sistemi a 64bit) o 4 byte(sistemi a 32 bit)



