TALLER PROGCOMP: TRACK BÁSICO

ARREGLOS

Gabriel Carmona Tabja

Universidad Técnica Federico Santa María,
Università di Pisa

April 16, 2024

Part I

ARREGLOS

ARREGLOS

Definición

Simple estructura de dato que:

- nos permitirá almacenar un único tipo de dato.
- tiene tamaño definido (tiempo de compilación o ejecución)
- los elementos estarán de manera **CONTIGUA** en la memoria

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

Tipos de Arreglos

- Arreglos Estáticos
  - Tamaño fijo
- Arreglos Dinámicos
  - Tamaño variable

Part II

Arreglos Estáticos

Arreglos Estáticos - Tamaño fijado en compilación

```cpp
int main() {
  int arr_init_values[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

  int arr[10];

  for(int i = 0; i < 10; i++) {
    cin >> arr[i];
  }

  for(int i = 0; i < 10; i++) {
    cout << arr[i] << " ";
  }
  cout << "\n";
}
```

Arreglos Estáticos - Tamaño fijado en ejecución

```cpp
int main() {
  int n;
  cin >> n;
  int arr[n];

  for(int i = 0; i < n; i++) {
    cin >> arr[i];
  }

  for(int i = 0; i < n; i++) {
    cout << arr[i] << " ";
  }
  cout << "\n";
  return 0;
}
```

Arreglos Estáticos - ¿Qué se puede almacenar?

```
1  struct Punto {/* codigo */};
2
3  int main() {
4    int arr[10];
5
6    char arr[10];
7
8    punto arr[10]
9
10   int matriz[10][10];
11
12   return 0;
13 }
```

Arreglos Estáticos - Complejidades

- Acceso a una posición:

Arreglos Estáticos - Complejidades

- Acceso a una posición: $O(1)$
- Actualización:

Arreglos Estáticos - Complejidades

- Acceso a una posición: $O(1)$
- Actualización: $O(1)$
- Agregar elementos: no se puede
- Eliminar elementos: tampoco :(

Part III

Arreglos Dinámicos a.k.a Vectores

VECTORES - OPERACIONES

- Acceder
- Agregar al final
- Eliminar del final
- Agregar en cualquier posición
- Eliminar de cualquier posición

VECTORES - AGREGAR AL FINAL

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

VECTORES - AGREGAR AL FINAL

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

Agregamos al final el valor 10

Arreglo	4	2	3	5	8	2	10
Pos	0	1	2	3	4	5	6

VECTORES - AGREGAR AL FINAL

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

Agregamos al final el valor 10

Arreglo	4	2	3	5	8	2	10
Pos	0	1	2	3	4	5	6

¿Cuál sería la complejidad?

VECTORES - AGREGAR AL FINAL

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

Agregamos al final el valor 10

Arreglo	4	2	3	5	8	2	10
Pos	0	1	2	3	4	5	6

¿Cuál sería la complejidad? $O(1)$

VECTORES - ELIMINAR DEL FINAL

Arreglo	4	2	3	5	8	2	10
Pos	0	1	2	3	4	5	6

Sacamos el último elemento.

VECTORES - ELIMINAR DEL FINAL

Arreglo	4	2	3	5	8	2	10
Pos	0	1	2	3	4	5	6

Sacamos el último elemento.

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

VECTORES - ELIMINAR DEL FINAL

Arreglo	4	2	3	5	8	2	10
Pos	0	1	2	3	4	5	6

Sacamos el último elemento.

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

¿Cuál sería la complejidad?

VECTORES - ELIMINAR DEL FINAL

Arreglo	4	2	3	5	8	2	10
Pos	0	1	2	3	4	5	6

Sacamos el último elemento.

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

¿Cuál sería la complejidad? $O(1)$

VECTORES - AGREGAR EN CUALQUIER POSICIÓN

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

Agreguemos el valor 10 en la posición 3.

VECTORES - AGREGAR EN CUALQUIER POSICIÓN

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

Agreguemos el valor 10 en la posición 3.

Arreglo	4	2	3	10	5	8	2
Pos	0	1	2	3	4	5	6

VECTORES - AGREGAR EN CUALQUIER POSICIÓN

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

Agreguemos el valor 10 en la posición 3.

Arreglo	4	2	3	10	5	8	2
Pos	0	1	2	3	4	5	6

¿Cuál sería la complejidad?

VECTORES - AGREGAR EN CUALQUIER POSICIÓN

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

Agreguemos el valor 10 en la posición 3.

Arreglo	4	2	3	10	5	8	2
Pos	0	1	2	3	4	5	6

¿Cuál sería la complejidad? $O(n)$

VECTORES - ELIMINAR EN CUALQUIER POSICIÓN

Arreglo	4	2	3	10	5	8	2
Pos	0	1	2	3	4	5	6

Eliminemos el elemento en la posición 3.

VECTORES - ELIMINAR EN CUALQUIER POSICIÓN

Arreglo	4	2	3	10	5	8	2
Pos	0	1	2	3	4	5	6

Eliminemos el elemento en la posición 3.

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

VECTORES - ELIMINAR EN CUALQUIER POSICIÓN

Arreglo	4	2	3	10	5	8	2
Pos	0	1	2	3	4	5	6

Eliminemos el elemento en la posición 3.

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

¿Cuál sería la complejidad?

VECTORES - ELIMINAR EN CUALQUIER POSICIÓN

Arreglo	4	2	3	10	5	8	2
Pos	0	1	2	3	4	5	6

Eliminemos el elemento en la posición 3.

Arreglo	4	2	3	5	8	2
Pos	0	1	2	3	4	5

¿Cuál sería la complejidad? $O(n)$

Vectores - Declaración

```cpp
int main() {
  vector< int > nums_1;

  int n;
  cin >> n;
  vector< int > nums_2(n);

  for(int i = 0; i < n; i++) {
    nums_2[i] = i;
  }

  vector< int > nums_3(n, 0);

  vector< char > chars(n);

  vector< vector< int > > matrix_1(n);

  vector< vector< int > > matrix_2(n, vector< int >(n));

  return 0;
}
```

VECTORES - PUSH_BACK Y POP_BACK

```cpp
int main() {
  vector< int > nums;

  int n;
  cin >> n;
  for(int i = 0; i < n; i++) {
    nums.push_back(i);
  }

  cout << nums.size() << "\n";

  while(nums.size()) {
    cout << nums[nums.size() - 1] << "\n";
    nums.pop_back();
  }

  cout << nums.size() << "\n";
  return 0;
}
```

VECTORES - INSERT Y ERASE

```cpp
int main() {
  int n;
  cin >> n;
  vector< int > nums(n, 10);

  cout << nums.size() << "\n";

  nums.insert(nums.begin() + 2, 9);

  cout << nums.size() << "\n";
  for(int i = 0; i < nums.size(); i++) {
    cout << nums[i] << "\n";
  }

  nums.erase(nums.begin() + 2);

  cout << nums.size() << "\n";
  for(int i = 0; i < nums.size(); i++) {
    cout << nums[i] << "\n";
  }
  return 0;
}
```

Resumen

- Arreglos:
  - Acceso $O(1)$
  - Útil cuando conoces el máximo
- Vectores:
  - Acceso $O(1)$
  - Agregar/Eliminar del final: $O(1)$
  - Agregar/Eliminar en cualquire posición: $O(n)$
  - Útil para algoritmos que requieran tamaño dinámico

REFERENCES I