

Tutoría 6: ordenando nombres

Defina un arreglo global que contenga nombres de la siguiente manera:

```
char nombres[][15] = {"Pedro", "Pablo", "Diego",  
"Daniel"};
```

Pasar nombres a minúsculas:

Utilizado la función `tolower` en el `main` de su programa, convierta todos los nombres del arreglo en minúsculas.

Heap:

Cree las funciones `int parent(int i)`, `int left(int i)` y `int right(int i)`. Estas funciones retornan `floor(i/2)`, `2*i` y `2*i+1` respectivamente.

Max-Heapify:

Defina la función:

```
void max_heapify(char ** A, int l, int i, int pos);
```

Esta función ayudará a convertir un arreglo `A` de nombres en un max heap. Donde `l` es el largo del sub arreglo que se quiere modificar, `A[i]` es un elemento que no cumple con la propiedad de max-heap y `pos` es la posición del carácter clave que se está ordenado. Debe asumirse que que los árboles binarios con raíces `left(i)` y `right(i)` son max-heaps.

Build-Max-Heap:

Defina la función:

```
void build_max_heap(char ** A, int l, int pos);
```

Esta función convierte un arreglo `A` en un max heap. Donde `l` es el largo del sub arreglo que se quiere modificar y `pos` es la posición del carácter clave que se está ordenado.

Heapsort:

Defina la función:

```
void heapsort(char ** A, int l, int pos);
```

Esta función ordena un arreglo `A`. Donde `l` es el largo del sub arreglo que se quiere ordenar y `pos` es la posición del carácter clave que se está ordenado.

Ordenando los nombres:

Primero intente ordenar los nombres alfabéticamente utilizando la función `heapsort`. Solo es necesario ordenar según el primer carácter.

Luego cree una función que ordene los nombres alfabéticamente variando los parámetros `A`, `l` y `pos`. La función debe ordenar primero los nombres según el primer carácter (`pos=0`), luego según el segundo (`pos=1`) hasta llegar a `pos=14`. Por ejemplo, con los nombres usados arriba ("Pedro", "Pablo", "Diego", "Daniel").

En la primera iteración se usaría `A=nombres`, `l=4` y `pos=0`. El orden resultante sería: ("Diego", "Daniel", "Pedro", "Pablo").

En la segunda iteración se usaría `A=nombres`, `l=2` y `pos=1`. El orden resultante sería: ("Daniel", "Diego", "Pedro", "Pablo").

En la tercera iteración se usaría `A=nombres+2`, `l=2` y `pos=1`. El orden resultante sería: ("Daniel", "Diego", "Pablo", "Pedro").