Nội dung

* Thuật toán sắp xếp cơ bản: sắp xếp nổi bot và lựa chọn

Ví dụ: So sánh thời gian thực hiện của 2 thuật toán nổi bọt và lựa chọn trên mảng

* Sinh ngẫu nhiên mảng 10000, 100000 phần tử
* So sánh thời gian thực hiện của 2 thuật toán

|  |
| --- |
| /\*  test thơi gian thục hien cua cac thuật toan co ban    \*\*/    #include<stdio.h>  #include<stdlib.h>  #include<time.h>  #include<string.h> // cho memcpy  // sinh mang ngau nhien  int \* generateArray(int size)  {  srand(time(NULL));  int \*retArr = (int\*)malloc(size\*sizeof(int));  for(int i=0; i<size; i++)  retArr[i]=rand();  return retArr;  }  // tao ra copy cua mang ban dau  int \*cloneArr(int \* input, int size)  {  int \*copyArr = (int\*)malloc(size\*sizeof(int));  memcpy(copyArr, input, size\*sizeof(int));  return copyArr;  }  // doi cho 2 phan tu  void swap(int \*a, int \*b)  {  int c = \*a;  \*a = \*b;  \*b = c;  }  // ham in mang  void printArr(const int \*input, int size)  {  printf("Cac phan tu trong mang kich thuoc %d la: \n",size);  for(int i=0; i<size; i++)  {  printf("%5d, ", input[i]);  if((i+1)%10==0) printf("\n");  }  printf("\n");  }  // sap xep noi bot  void bubblesort(int \*input, int size)  {  for(int i=size; i>1; i--)  for(int j=1; j<i; j++)  if(input[j-1]>input[j]) swap(&input[j-1], &input[j]);  }  // sap xep lua chon  void selectionsort(int \*input, int size)  {  for(int i=size; i>1; i--) // cuoi day la i-1  {  int vtMax = 0;  for(int j=1; j<i; j++)  if(input[j]>input[vtMax]) vtMax=j;  swap(&input[vtMax], &input[i-1]);  }  }    int main()  {  int \*A, size;  size = 100;  A = generateArray(size);  printf("Mang ban dau: \n");  printArr(A, size);  // minh hoa bubble sort  int \*copiedArr = cloneArr(A, size);  bubblesort(copiedArr, size);  printf("Mang sau khi bubblesort: \n");  printArr(copiedArr, size);    // minh hoa selection sort  free(copiedArr);  copiedArr = cloneArr(A, size);  selectionsort(copiedArr, size);  printf("Mang sau khi selectionsort: \n");  printArr(copiedArr, size);  return 0;  } |

**~~Bài tập 1~~**~~. So sánh thời gian thực hiện của 2 thuật toán trên khi sắp xếp trên dãy~~

* 1000
* 10000
* Và 100000 phần tử

Giải thích vì sao cần free(copiedArr);

|  |
| --- |
| /\*  test thơi gian thục hien cua cac thuật toan co ban    \*\*/    #include<stdio.h>  #include<stdlib.h>  #include<time.h>  #include<string.h> // cho memcpy  // sinh mang ngau nhien  int \* generateArray(int size)  {  srand(time(NULL));  int \*retArr = (int\*)malloc(size\*sizeof(int));  for(int i=0; i<size; i++)  retArr[i]=rand();  return retArr;  }  // tao ra copy cua mang ban dau  int \*cloneArr(int \* input, int size)  {  int \*copyArr = (int\*)malloc(size\*sizeof(int));  memcpy(copyArr, input, size\*sizeof(int));  return copyArr;  }  // doi cho 2 phan tu  void swap(int \*a, int \*b)  {  int c = \*a;  \*a = \*b;  \*b = c;  }  // ham in mang  void printArr(const int \*input, int size)  {  printf("Cac phan tu trong mang kich thuoc %d la: \n",size);  for(int i=0; i<size; i++)  {  printf("%5d, ", input[i]);  if((i+1)%10==0) printf("\n");  }  printf("\n");  }  // sap xep noi bot  void bubblesort(int \*input, int size)  {  for(int i=size; i>1; i--)  for(int j=1; j<i; j++)  if(input[j-1]>input[j]) swap(&input[j-1], &input[j]);  }  // sap xep lua chon  void selectionsort(int \*input, int size)  {  for(int i=size; i>1; i--) // cuoi day la i-1  {  int vtMax = 0;  for(int j=1; j<i; j++)  if(input[j]>input[vtMax]) vtMax=j;  swap(&input[vtMax], &input[i-1]);  }  }  // sap xep chen tren mang  void insertionSort(int \*A, int N) {  // index tu 1 -> N  for(int k = 2; k <= N; k++)  {  int last = A[k];  int j = k;  while(j > 1 && A[j-1] > last)  {  A[j] = A[j-1];  j--;  }  A[j] = last;  }  }    int main()  {  int \*A, size;  size = 100;  A = generateArray(size);  printf("Mang ban dau: \n");  printArr(A, size);  // minh hoa bubble sort  int \*copiedArr = cloneArr(A, size);  bubblesort(copiedArr, size);  printf("Mang sau khi bubblesort: \n");  printArr(copiedArr, size);    // minh hoa selection sort  free(copiedArr);  copiedArr = cloneArr(A, size);  selectionsort(copiedArr, size);  printf("Mang sau khi selectionsort: \n");  printArr(copiedArr, size);  // minh hoa insertion sort  free(copiedArr);  copiedArr = cloneArr(A, size);  insertionSort(copiedArr, size);  printf("Mang sau khi insertionsort: \n");  printArr(copiedArr, size);  return 0;  } |

**Bài tập 2**. Cho danh sách n số nguyên được sinh ngẫu nhiên KHÔNG trùng nhau. Ta có thể tóm tắt dãy nếu các phần tử trong dãy là liên tiếp như sau

Danh sách ban đầu: 1,3,6,5,4,7,8, 10,12

Sau khi sắp xếp lại: 1,**3,4,5,6,7,8**,10,12

Tóm tắt: 1,3-8,10,12

Khi tóm tắt các phần tử liên tiếp có thể được viết gọn lại (nếu chỉ có 2 số thfi có thể viện gọn hoặc không. VD. 3,4 hoặc 3-4)

Yêu cầu

* Sinh ngẫu nhiên n (n<=M) phần tử KHÔNG trùng nhau trong khoảng 0-M (M=1000, 10000, 50000)
* Sắp xếp dãy
* Tóm tắt lại dãy

**Bài tập 3**. Cần vận chuyển n kiện hàng có khối lượng lần lượt w1---> wn (Khối lượng có thể sinh ngẫu nhiên). Giả sử bạn có thể thuê được các xe tải vận chuyển với khối lượng chuyển chở tối đa là M.

Hỏi liệu có thể vận chuyển được n kiện hàng trên hay không, và nếu vận chuyển được thì cần thuê tối thiểu bao nhiêu xe?

VD 1. có 5 kiện hàng khối lượng lần lượt 12, 3, 17, 500, 4

Và xe tải chỉ có thể vận chuyển tối đa M = 300

Vậy ta KHÔNG thể chuyển 5 kiện hàng đó

VD 2. có 5 kiện hàng khối lượng lần lượt 12, 30, 17, 50, 4

Và xe tải chỉ có thể vận chuyển tối đa M = 100

Thì ta có thể vận chuyển được và phải thuê tối thiểu 2 xe tải

VD 3. có 5 kiện hàng khối lượng lần lượt 25, 30, 17, 50, 40

Và xe tải chỉ có thể vận chuyển tối đa M = 50

Thì ta có thể vận chuyển được và phải thuê tối thiểu 4 xe tải