Nội dung. Đánh giá các thuật toán sắp xếp nâng cao

* Sắp xếp trộn
* Sắp xếp nhanh
* Sắp xếp vun đống

Ví dụ 1. So sánh thời gian thực hiện của các thuật toán sắp xếp cơ bản sau trên mảng (6 thuật toán)

|  |
| --- |
| /\*  test thoi gian th?c hien cua cac thu?t toan co ban    \*\*/    #include<stdio.h>  #include<stdlib.h>  #include<time.h>  #include<string.h> // cho memcpy  #include <stdint.h>  #include <sys/time.h>    // ham tru thoi gian  struct timeval timeSub(struct timeval s, struct timeval e)  {  struct timeval tval\_result;  tval\_result.tv\_sec = e.tv\_sec - s.tv\_sec;  tval\_result.tv\_usec = e.tv\_usec - s.tv\_usec;  if(tval\_result.tv\_usec<0) {  tval\_result.tv\_usec = 1000000+tval\_result.tv\_usec;  tval\_result.tv\_sec--;  }  return tval\_result;  }    // 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;  }  }    void merge(int arr[], int l, int m, int r) {  int i, j, k;  int n1 = m - l + 1;  int n2 = r - m;    int L[n1], R[n2];    for (i = 0; i < n1; i++)  L[i] = arr[l + i];  for (j = 0; j < n2; j++)  R[j] = arr[m + 1 + j];    i = 0;  j = 0;  k = l;  while (i < n1 && j < n2) {  if (L[i] <= R[j]) {  arr[k] = L[i];  i++;  } else {  arr[k] = R[j];  j++;  }  k++;  }    while (i < n1) {  arr[k] = L[i];  i++;  k++;  }    while (j < n2) {  arr[k] = R[j];  j++;  k++;  }  }    void mergeSort(int arr[], int l, int r) {  if (l < r) {  int m = l + (r - l) / 2;    mergeSort(arr, l, m);  mergeSort(arr, m + 1, r);    merge(arr, l, m, r);  }  }      int partition (int arr[], int low, int high)  {  int pivot = arr[high];  int i = (low - 1);    for (int j = low; j <= high- 1; j++)  {  if (arr[j] <= pivot)  {  i++;  swap(&arr[i], &arr[j]);  }  }  swap(&arr[i + 1], &arr[high]);  return (i + 1);  }    void quickSort(int arr[], int low, int high)  {  if (low < high)  {  int pi = partition(arr, low, high);  quickSort(arr, low, pi - 1);  quickSort(arr, pi + 1, high);  }  }    void heapify(int arr[], int n, int i)  {  int largest = i;  int left = 2 \* i + 1;  int right = 2 \* i + 2;  if (left < n && arr[left] > arr[largest])  largest = left;  if (right < n && arr[right] > arr[largest])  largest = right;  if (largest != i)  {  int temp = arr[i];  arr[i] = arr[largest];  arr[largest] = temp;  heapify(arr, n, largest);  }  }    void heapSort(int arr[], int n)  {  for (int i = n / 2 - 1; i >= 0; i--)  heapify(arr, n, i);  for (int i = n - 1; i >= 0; i--)  {  int temp = arr[0];  arr[0] = arr[i];  arr[i] = temp;  heapify(arr, i, 0);  }  }    // shellsort dua tren thuat toan sap xep lua chon  void shellSort(int arr[], int n)  {  for (int gap = n/2; gap > 0; gap /= 2)  {  for (int i = gap; i < n; i++)  {  int temp = arr[i];  int j;  for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)  arr[j] = arr[j - gap];  arr[j] = temp;  }  }  }        int main()  {  int \*A, size;  size = 100000;  A = generateArray(size);  printf("Mang ban dau: \n");  //printArr(A, size);  // minh hoa bubble sort  int \*copiedArr = cloneArr(A, size);  // tinh thoi gian  printf("Sap xep noi bot:\n");  struct timeval tval\_before, tval\_after, tval\_result;  gettimeofday(&tval\_before, NULL);  bubblesort(copiedArr, size);  gettimeofday(&tval\_after, NULL);  tval\_result = timeSub(tval\_before, tval\_after);  printf("thoi gian sap xep : %ld.%06ld\n", (long int)tval\_result.tv\_sec, (long int)tval\_result.tv\_usec);  //printf("Mang sau khi bubblesort: \n");  //printArr(copiedArr, size);      // minh hoa selection sort  printf("Sap xep lua chon:\n");  free(copiedArr);  copiedArr = cloneArr(A, size);  gettimeofday(&tval\_before, NULL);  selectionsort(copiedArr, size);  gettimeofday(&tval\_after, NULL);  tval\_result = timeSub(tval\_before, tval\_after);  printf("thoi gian sap xep : %ld.%06ld\n", (long int)tval\_result.tv\_sec, (long int)tval\_result.tv\_usec);  //printf("Mang sau khi selectionsort: \n");  //printArr(copiedArr, size);  // minh hoa insertion sort  printf("Sap xep chen - Insertion Sort:\n");  free(copiedArr);  copiedArr = cloneArr(A, size);  gettimeofday(&tval\_before, NULL);  insertionSort(copiedArr, size);  gettimeofday(&tval\_after, NULL);  tval\_result = timeSub(tval\_before, tval\_after);  printf("thoi gian sap xep : %ld.%06ld\n", (long int)tval\_result.tv\_sec, (long int)tval\_result.tv\_usec);  //printf("Mang sau khi insertionsort: \n");  //printArr(copiedArr, size);    // Sap xep Shellsort - Shellsort  printf("Sap xep Shellsort - Shellsort:\n");  free(copiedArr);  copiedArr = cloneArr(A, size);  gettimeofday(&tval\_before, NULL);  shellSort(copiedArr,size);  gettimeofday(&tval\_after, NULL);  tval\_result = timeSub(tval\_before, tval\_after);  printf("thoi gian sap xep : %ld.%06ld\n", (long int)tval\_result.tv\_sec, (long int)tval\_result.tv\_usec);  // Sap xep tron - MergeSort  printf("Sap xep tron - Merge Sort:\n");  free(copiedArr);  copiedArr = cloneArr(A, size);  gettimeofday(&tval\_before, NULL);  mergeSort(copiedArr,0, size);  gettimeofday(&tval\_after, NULL);  tval\_result = timeSub(tval\_before, tval\_after);  printf("thoi gian sap xep : %ld.%06ld\n", (long int)tval\_result.tv\_sec, (long int)tval\_result.tv\_usec);  // sap xep nhanh - Quick sort  printf("Sap xep nhanh - Quick sort:\n");  free(copiedArr);  copiedArr = cloneArr(A, size);  gettimeofday(&tval\_before, NULL);  quickSort(copiedArr,0, size);  gettimeofday(&tval\_after, NULL);  tval\_result = timeSub(tval\_before, tval\_after);  printf("thoi gian sap xep : %ld.%06ld\n", (long int)tval\_result.tv\_sec, (long int)tval\_result.tv\_usec);  // sap xep vun dong - Heapsort  printf("Sap xep vun dong - Heapsort:\n");  free(copiedArr);  copiedArr = cloneArr(A, size);  gettimeofday(&tval\_before, NULL);  heapSort(copiedArr,size);  gettimeofday(&tval\_after, NULL);  tval\_result = timeSub(tval\_before, tval\_after);  printf("thoi gian sap xep : %ld.%06ld\n", (long int)tval\_result.tv\_sec, (long int)tval\_result.tv\_usec);  return 0;  } |

**Bài tập 1**. Trong một cuộc chạy marathon mở rộng cho n người, ban tổ chức muốn thống kê chung về thời gian chạy và top người nhanh nhất. Mỗi người tham gia sau khi cung cấp họ tên sẽ được cung cấp 1 mã số tham dự và được ghi nhận thời gian xuất phát (chạy qua vạch xuất phát) và thời gian về đích( vượt qua vạch đích).

Thông tin được cho dưới dạng file như sau

|  |
| --- |
| Nguyen Van A:1  09452 08:20:01 10:02:23  Le Thi B:0  23233 08:20:02 10:05:23  Le Van C:1  23233 08:20:12 10:15:33  Le Van D:0  23233 08:20:12  Le Van E:1  23233 08:19:12 11:15:33  # |

Các dòng sẽ gồm

* Tên người
* Giới tính : 1 là Nam và 0 là nữ
* Mã số tham gia
* Thời gian bắt đầu
* Thời gian về đích (nếu hoàn thành, nếu không hoàn thành thì thời gian này sẽ bỏ trống)

Thời gian chạy và kết thúc là trong ngày

Kết thúc sẽ là dấu #

Hãy dựa vào thông tin nhà tổ chức cung cấp để tìm và đưa ra

* Thời gian chạy nhanh nhất
* Người/nhóm người chạy nhanh nhất
* Thời gian chạy trung bình của top 10, top 100
* Số lượng người tham gia, số lượng người hoàn thành cuộc đua
* Thời gian chạy cần chia theo giới tính nam/nữ

**Bài tập 2**. Cho 1 đoạn văn bản tiếng anh, nhập từ file. Ta coi mỗi từ sẽ được ngăn cách bởi dấu câu, hoặc dấu cách trống. Hãy thống kê và đưa ra

* Tần xuất các từ dựa trên độ dài. VD. Từ có 2,3,4.... ký tự xuất hiện bao lần
* Từ dài nhất và từ ngắn nhất
* Tần số xuất hiện các ký tự. VD. Ký tự nào hay dùng nhất và ký tự nào ít dùng
* Câu dài nhất và câu ngắn nhất (gồm bao từ)

VD. Văn bản đầu vào

|  |
| --- |
| US Navy personnel have been removing items from boats and loading material onto trucks at the Johnny Causey Boat Landing in North Myrtle Beach, South Carolina on Sunday, according to a CNN team at the scene.  In cell phone video shot earlier in the day that was obtained by CNN, a pile of white material was visible on the deck of one of the boats and several people dressed in camouflage could be seen near the boat.  Another boat that appeared to hold similar material could also be seen at a nearby dock. The people could also be seen unloading several boxes off one of the boats.  CNN cannot confirm that the material is debris from the suspected Chinese spy balloon.  On Sunday night, the personnel would not say what they were doing or why they were working at the boat landing, which is situated along the Intercoastal Waterway in North Myrtle Beach.  They have been identified as Navy personnel according to their uniforms and vessel signage present at the location.  CNN has reached out the Navy Sunday afternoon. |