

Ishod učenja I5

Zadatak 51. Na raspolaganju su vam Selection sort i Merge sort (u priložima). Koristeći podatke sa slike 1, od ta dva algoritma odaberite onaj koji ima bolju složenost u najgorem slučaju. Promijenite odabrani algoritam tako da sortira pravokutnike (širina, visina) rastuće prema površini. Učitajte svih 1000 pravokutnika iz datoteke pravokutnici.txt (u svakom retku su širina i visina jednog pravokutnika odvojeni razmakom) i prepisite ih u drugu datoteku, ali sortirano rastuće.

Algorithm	Time Complexity		
	Best	Average	Worst
Quicksort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$
Mergesort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$
Timsort	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$
Heapsort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$
Bubble Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$
Insertion Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$
Selection Sort	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$
Tree Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$
Shell Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))^2$	$O(n \log(n))^2$
Bucket Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$
Radix Sort	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$
Counting Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n+k)$
Cubesort	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$

Slika 1

Zadatak 52. Generirajte vektor od razbacanih brojeva od 1 do 100. Sortirajte ga na sljedeća tri načina te rezultate svakog sortiranja zapišite u posebnu datoteku (pokretanjem programa trebaju nastati tri datoteke):

- Rastuće
- Padajuće
- Tako da prvo dođu parni, a onda neparni brojevi.

Zadatak 53. Napišite program koji sortira riječi iz datoteke osobe.txt na način da prvo dođu dulje, a onda kraće riječi. Riječi jednake duljine neka budu sortirane abecedno.