

Question 1: Sammenlikningsbasert sortering



Hva er worst case for sammenlikningsbasert sortering? Velg det alternativet som passer best.

- ☒ $\Omega(n \log n)$
- ☐ $\Theta(n \log n)$
- ☐ $\Theta(n^2)$
- ☐ $O(n^2)$
- ☐ $\Theta(n)$
- ☐ $O(n)$
- ☐ $O(n \log n)$
- ☐ $\Omega(n)$
- ☐ $\Omega(n^2)$

Question 2: Sammenlikningsbasert sortering

Insertion sort har kjøretid $\Theta(n^2)$ i verste tilfelle.

Insertion sort har kjøretid $\Theta(n)$ i best case (når tabellen allerede er sortert).



Anta at du har en sammenlikningsbasert sorteringsalgoritme S som sorterer heltallene i en tabell med lengde n . Hvilke(t) alternativ er korrekt?

- ☒ S kan ha best case-kjøretid $\Theta(n)$.
- ☒ S kan ha worst case-kjøretid $\Theta(n^2)$.
- ☐ S kan telle forekomster i tabellen for å oppnå kjøretid på $\Theta(n)$.
- ☐ S kan være bucket sort.

Question 3: Stabil sortering

Anta følgende tabell med elementer: $[(x: 3, y: 5), (x: 1, y: 5), (x: 3, y: 3), (x: 3, y: 4)]$

Dersom en stabil sorteringsalgoritme brukes for å sortere tabellen på x-verdiene, hvordan vil listen se ut sortert?

- ☐ $[(x: 3, y: 5), (x: 3, y: 3), (x: 3, y: 4), (x: 1, y: 5)]$
- ☐ $[(x: 1, y: 5), (x: 3, y: 3), (x: 3, y: 4), (x: 3, y: 5)]$
- ☒ $[(x: 1, y: 5), (x: 3, y: 5), (x: 3, y: 3), (x: 3, y: 4)]$
- ☐ $[(x: 3, y: 3), (x: 3, y: 4), (x: 3, y: 5), (x: 1, y: 5)]$

Question 4: Stabil sortering

Hvilke(n) av de følgende algoritmene er stabil(e), gitt implementasjonen i pensumboka?

- ☒ Counting sort
- ☒ Insertion sort
- ☒ Mergesort
- ☐ Quicksort
- ☒ Radix sort
- ☐ Heapsort
- ☒ Bubblesort

Question 5: Counting sort

Anta at du skal bruke counting sort for å sortere n heltall, der hvert heltall er mellom 0 og k . Du vet at k er mye større enn n . Hva er den mest presise beskrivelsen av kjøretiden?

- ☐ Ingen av alternativene.
- ☒ $\Theta(k)$
- ☐ $\Theta(n)$
- ☐ $\Theta(n + k)$

Question 6: Counting sort



Hvorfor itererer counting sort bakover gjennom A når heltallene skal settes inn i B?

- ☒ Fordi algoritmen skal være stabil.
- ☐ Fordi vi får elementer i tilfeldig rekkefølge om vi går forover gjennom A.
- ☐ Fordi vi ønsker å sette inn elementer som skal sist i B først.
- ☐ Fordi algoritmen krasjer om vi går forover gjennom A.

Question 7: Radix sort



Radix sort sorterer på det mest signifikante sifferet først.

- ☒ Usant.
- ☐ Sant.

Question 8: Radix sort

Insertion sort går tregere enn mergesort.



Hvilken av disse algoritmene ville vært den beste som en del av radix sort?

- ☐ Quicksort
- ☒ Insertion sort
- ☐ Mergesort
- ☐ Bucket sort

Question 9: Radix sort



Du skal sortere n heltall med d siffer ved hjelp av radix sort. Du kan anta at counting sort brukes som sorteringsrutine og at $k = \Theta(\log n)$. Hva er den mest presise beskrivelsen av kjøretiden til algoritmen?

- ☒ $\Theta(dn)$
- ☐ $\Theta(n + k)$
- ☐ $\Theta(n \log n)$
- ☐ $\Theta(d + n)$

Question 10: Radix sort



Det er mulig å sortere desimaltall med to desimaler ved hjelp av radix sort.

- ☐ Usant.
- ☒ Sant.

Question 11: Bucket sort



Bucket sort har en worst case-kjøretid på $\Theta(n)$.

- ☒ Usant.
- ☐ Sant.

Question 12: Bucket sort



Hva er kjøretiden for bucket sort i average case om vi bruker tre bøtter?

- ☐ $O(n/3)$
- ☐ $O(3n)$
- ☐ $O(n)$
- ☒ $O(n^2)$

Question 13: Median



Hva er medianen i en sortert tabell?

- ☐ Gjennomsnittet.
- ☐ Det tallet som forekommer oftest.
- ☐ Differansen mellom største og minste tall.
- ☒ Det midterste elementet.

Question 14: Median



Vi kan ikke finne det k -te største elementet til en usortert liste uten å sortere den.

- ☒ Usant.
- ☐ Sant.

Question 15: Randomized Select



Hva er forventet kjøretid for randomized select?

- ☐ $\Theta(n^2)$
- ☐ $\Theta(n \log n)$
- ☐ $\Theta(\log n)$
- ☒ $\Theta(n)$

Question 16: Randomized select



Randomized select har bedre worst case-kjøretid enn randomized quicksort.

- ☐ Sant.
- ☒ Usant.

Question 17: Select



Hva er worst case-kjøretiden til select?

- ☐ $\Theta(n \log n)$
- ☐ $\Theta(n^2)$
- ☐ Ingen av alternativene.
- ☒ $O(n)$