

Provkonstruktion

Årskurs: Gymnasiet

Ämne: Tillämpad programmering

Tema: Datastrukturer och algoritmer

Syfte

Syftet med provet är att bedöma studenternas kunskaper och färdigheter i att förstå och använda olika datastrukturer och algoritmer, samt deras förmåga att lösa problem relaterade till dessa. Provets frågor syftar till att både testa faktakunskap och ge möjlighet för djupare analys och resonemang kring programmeringens grundläggande koncept.

Koppling till styrdokument

Centralt innehåll

Denna lektion introducerar eleverna till grundläggande datastrukturer såsom arrayer, listor och stackar samt algoritmer för sortering och sökning. Eleverna kommer att lära sig hur dessa datastrukturer används för att lagra och manipulera data effektivt.

Kunskapskrav

Eleven ska kunna använda olika datastrukturer och algoritmer för att lösa problem samt förstå och analysera olika algoritmers effektivitet.

Prov

Faktafrågor

1. Vad är en datastruktur?
 - A. En typ av programmeringsspråk
 - B. En metod för att debugg'a kod
 - C. ****Ett sätt att organisera och lagra data så att den kan användas effektivt.****
 - D. En typ av algoritm
2. Vilken av följande är en typ av datastruktur?
 - A. ****Arrayer****
 - B. Variabler
 - C. Funktioner
 - D. Klasser

3. Vad gör en stack?

- A. Den lagrar data i en FIFO-ordning - först in, först ut.
- B. ****Den lagrar data i en LIFO-ordning - senast in, först ut.****
- C. Den är alltid sorterad.
- D. Den kan inte lagra data.

4. Vilken sorteringsalgoritm är enklast?

- A. Snabbsortering
- B. ****Bubbsortering****
- C. Merge-sortering
- D. Insertion-sortering

5. Vad betyder tidskomplexitet?

- A. Hur snabbt en dator kör program
- B. ****Ett mått på hur mycket tid en algoritm tar att köra baserat på storleken av ingångsdata.****
- C. Hur mycket minne en program använder
- D. Ett sätt att mäta programfel

6. Vilken av följande operationer används för att lägga till ett element i en stack?

- A. Pop
- B. ****Push****
- C. Shoot
- D. Pull

7. Vad är en array?

- A. En typ av loop
- B. ****En datastruktur som lagrar flera värden i en sekventiell ordning.****
- C. En typ av funktion
- D. En algoritm för att sortera data

8. Vilken av följande algoritmer används för sökning?

- A. ****Binärsökning****
- B. Bubbsortering
- C. Merge-sortering
- D. Quick-sortering

9. Vad är skillnaden mellan arrayer och listor?

- A. Listor är alltid sorterade
- B. Arrayer kan inte ändras
- C. ****Listor kan ändra storlek dynamiskt medan arrayer har en fast storlek.****
- D. Det finns ingen skillnad

10. Vilken datastruktur använder LIFO-principen?

- A. ****Stack****

- B. Kö
- C. Array
- D. Lista

11. I vilken situation är det bra att använda en array?

- A. När mängden data är dynamisk
- B. **När man vet exakt hur många element som ska lagras i förväg.**
- C. När man behöver snabb åtkomst till element
- D. När man inte behöver manipulera datan

12. Vilken av följande är en egenskap hos en sorteringsalgoritm?

- A. Den kan alltid garantera en optimal lösning
- B. **Den sorterar data i en viss ordning.**
- C. Den måste alltid använda arrayer
- D. Den kan inte manipuleras

13. Vilken tidskomplexitet har bubbsortering i värsta fall?

- A. $O(n \log n)$
- B. **$O(n^2)$**
- C. $O(n)$
- D. $O(1)$

14. Vad är en lista?

- A. En typ av array
- B. **En datastruktur som kan innehålla olika datatyper och har en dynamisk storlek.**
- C. En algoritm för att organisera data
- D. En typ av funktion

15. Vilket av följande påståenden om stackar är sant?

- A. De tillåter både FIFO och LIFO
- B. **De tillåter endast LIFO-operationer.**
- C. De är inte användbara
- D. De kan inte implementeras med arrayer

Resonerande frågor

1. Förklara skillnaden mellan en lista och en array samt ge exempel på när det kan vara mer lämpligt att använda den ena framför den andra. Syftet är att ge eleverna möjlighet att visa djupare förståelse för de två datastrukturernas användningsområden och konsekvenser av deras val.

2. Analysera tidskomplexiteten för bubbsortering och diskutera varför det ofta anses vara en ineffektiv sorteringsalgoritm. Här får eleverna möjlighet att koppla tidskomplexitet till prestanda och ge insikter i algoritmers effektivitet.

3. Beskriv en situation i programutveckling där stackar skulle vara det mest lämpliga valet av datastruktur.

Denna fråga uppmanar eleverna att leta efter praktiska exempel på stackars användning i verkliga situationer.

4. Diskutera hur en algoritm som kombinerar sortering och sökning kan implementeras effektivt i en applikation.

Eleverna får möjlighet att visa sin förståelse för sambandet mellan datastrukturer och algoritmer och hur de kan integreras.

5. Ge exempel på hur stackar kan användas i recursive funktioner.

Denna fråga ger eleverna chansen att visa förståelse för hur stackar kan vara en grundläggande del av rekursion.

6. Jämför och kontrastera effektiviteten mellan olika sorteringsalgoritmer i olika scenarier.

Eleverna uppmanas att tänka kritiskt och ge en jämförande analys av algoritmerna.

7. Hur kan kunskap om datastrukturer förbättra programvaruarkitektur och design?

Syftet är att få eleverna att tänka på storleken av system och effektivitetsaspekter i arkitektur och design.

8. Reflektera över hur förståelse av datastrukturer och algoritmer kan påverka en programmerares yrkesutveckling.

Denna fråga ger eleverna chansen att tänka på den personliga och professionella betydelsen av sina kunskaper.

Bedömning

Faktafrågor: 1 poäng per fråga, maximalt 15 poäng.

Resonerande frågor: 3 poäng per fråga, maximalt 24 poäng.

För betyg E krävs 8 poäng, för betyg C krävs 12 poäng (varav minst 3 poäng från resonerande frågor), och för betyg A krävs 18 poäng (varav minst 5 poäng från resonerande frågor).

Tags: [Gymnasiet](#), [Programmering](#), [Tillämpad programmering](#)