

# Provkonstruktion

**Årskurs:** Gymnasiet

**Ämne eller kurs:** Dator teknik 1b

**Tema:** Algoritmer och datastrukturer

---

## Syfte

Syftet med provet är att bedöma elevernas kunskaper inom algoritmer och datastrukturer, samt deras förmåga att förstå och tillämpa dessa i programmeringslösningar.

## Koppling till styrdokument

### Centralt innehåll

Lektionens centrala innehåll omfattar förståelsen av grundläggande algoritmer och datastrukturer, hur man designar och implementerar dem effektivt samt deras tillämpningar i programmeringslösningar.

### Kunskapskrav

Eleven ska kunna beskriva och ge exempel på olika algoritmer och datastrukturer, samt kunna implementera och använda dessa i programmeringsuppgifter.

---

## Prov

### Faktafrågor

1. Vad är en algoritm?

- A) En programkod
- B) En datastruktur
- C) En **sekvens av steg för att lösa ett problem**
- D) En programmeringsspråk

2. Vilken av följande är en sorteringsalgoritm?

- A) Insertion sort
- B) Analys algoritm
- C) En **sorteringsmetod**
- D) Rekursiv algoritm

3. Vad är en array?

- A) En loop
- B) En **datastruktur som lagrar element i sekventiell ordning**
- C) En algoritm
- D) En funktion

4. Vilken datastruktur används för att lagra element i en LIFO-ordning?

- A) Kö
- B) Lista
- C) **Stack**
- D) Array

5. Vad beskriver tidskomplexitet?

- A) Hur mycket minne en algoritm använder
- B) Hastigheten på en dator
- C) **Hur lång tid en algoritm tar i förhållande till inputdata**
- D) Antal programmeringsspråk

6. Vilket av följande är inte en typ av datastruktur?

- A) Lista
- B) Stack
- C) Kö
- D) **Användargränssnitt**

7. Vad är Big O-notation?

- A) En typ av datastruktur
- B) En algoritm
- C) **En notationssystem för att beskriva algoritmers prestanda**
- D) En programmeringsspråk

8. Vad beskriver aktuell algoritmisk komplexitet?

- A) Antal iterationer
- B) **Tid och rum i förhållande till problemets storlek**
- C) Antal rader kod
- D) Algoritmens effektivitet

9. Vilken av följande är en egenskap hos en kö?

- A) Den är osorterad
- B) Den är **FIFO (First In First Out)**
- C) Den kan lagra element i slumpmässig ordning
- D) Den sparar endast unika värden

10. Hur implementeras en algoritm i ett programmeringsspråk?

- A) Genom att rita diagram
- B) **Genom att skriva kod**
- C) Genom att diskutera
- D) Genom att analysera tid

11. Vilken algoritm är bäst för att sortera en liten mängd data?

- A) Merge sort
- B) Quick sort
- C) **Bubbel-sortering**
- D) Heapsort

12. Vad är syftet med en datastruktur?

- A) Att lagra kommentarer
- B) **Att organisera data för effektiv hantering**
- C) Att variera kodsyntax
- D) Att få datorn att köra snabbare

13. Vilken av följande påståenden är sant för algoritmer?

- A) De måste vara långa
- B) **De måste vara tydliga och effektiva**
- C) De kan vara oorganiserade
- D) De är alltid lösningen

14. Vid vilken situation är det bäst att använda en lista snarare än en array?

- A) När element inte ska ändras
- B) **När man ofta lägger till och tar bort element**
- C) När man vill ha en fast storlek
- D) När man arbetar med stora datamängder

15. Hur analyserar man en algoritms effektivitet?

- A) Genom att räkna linjer
- B) Genom att jämföra med andra algoritmer
- C) **Genom att använda Big O-notation**

- D) Genom att kolla på minnesanvändning

## Resonerande frågor

1. Förklara vad en algoritm är och ge exempel på hur den kan användas i praktiska tillämpningar.

Syftet är att ge eleverna möjlighet att visa djup förståelse för algoritmer och deras praktiska betydelse.

2. Diskutera skillnaderna mellan olika datastrukturer och i vilka situationer det är mest fördelaktigt att använda dem.

Syftet är att uppmuntra eleverna att tänka kritiskt kring valet av datastruktur baserat på problemlösning.

3. Hur kan tidskomplexitet påverka valet av algoritm i ett program?

Syftet är att få eleverna att överväga effekterna av tidskomplexitet på prestanda i program.

4. Jämför och kontrastera bubbel-sortering med en effektivare sorteringsalgoritm såsom quicksort.

Syftet är att ge eleverna möjlighet att analysera och förstå skillnader mellan olika sorteringsalgoritmer.

5. Vilka faktorer bör man överväga när man designar en algoritm?

Syftet är att eleverna ska tänka på både effektivitet och tydlighet i algoritmdesign.

6. Beskriv ett praktiskt problem och hur en specifik algoritm skulle lösa det.

Syftet är att koppla teori till praktik och ge exempel på algoritmers tillämpningar.

7. Diskutera hur du skulle förklara Big O-notation för någon som är nybörjare inom programmering.

Syftet är att bedöma elevernas förmåga att förklara komplexa begrepp på ett enkelt sätt.

8. Hur skulle du göra en algoritm mer effektiv och varför skulle det vara viktigt?

Syftet är att få eleverna att tänka på algoritmers praktiska betydelse och hur små förändringar kan påverka effektivitet.

## Bedömning

Faktafrågorna ger 1 poäng per korrekt svar. Resonerande frågor bedöms med 2 poäng per fråga.

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

Tags: [Datorteknik 1b](#), [Gymnasiet](#)