Algoritmer och datastrukturer
Rosemarie Wolf

# Rapport

1.

Jag valde att spara uppgifterna från filen i en hashmap. Detta för att det då bara tar konstant tid att plocka ut varje enskilt mätvärde. Hade jag sparat det i ett träd hade tidskomplexiteten varit logaritmisk istället. Jag skapade en ny klass, Measurement, för att spara uppgifter om temperaturer och antal godkända värden för varje datum, i ett eget objekt. En hashmap var då perfekt för att spara ett objekt som element med ett datum som nyckel. Dessa objekt hade också kunnat sparas i en lista, men då hade det tagit längre tid (linjär) att traversera den för att plocka ut ett specifikt datum.

2.

Alla algoritmer är iterativa. Eftersom algoritmerna ska gå igenom ett datumintervall, och för varje datum plocka ut ett värde från ett measurement-objekt, så tycker jag att en iterativ lösning passar bäst. Jag har inte använt någon av designteknikerna.

3.

Alla algoritmer, bortsett från missingValues, har en komplexitet på O(n). Detta beror på att de itererar genom alla värden i ett datumintervall. Sedan plockar de ut värden för varje datum, men detta tar endast konstant tid då de är sparade i en hashmap. Algoritmen missingValues har en tidskomplexitet på O(n * logn). Detta beror på att den anropar javas funktion sort, som sorterar en lista. Det är alltså denna funktion som gör att denna algoritm får denna tidskomplexitet.