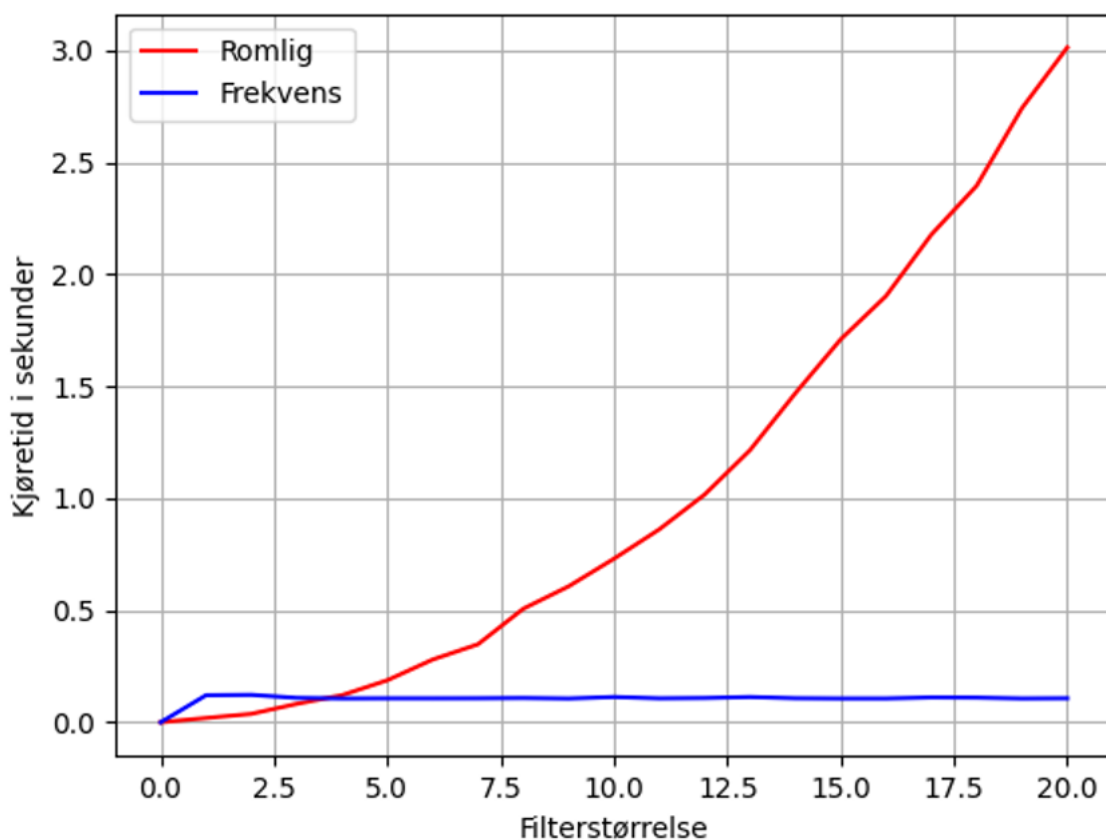


Oppgave 1.2

1. Forflytningen av det frekvsensdomenefiltrerte bildet skjer fordi ved filtrering i frekvsensdomenet må bildet også nullutvides for å transformere det til riktig størrelse. For å få riktig størrelse må man dermed trimme bildestørrelsen ned til originalen når man transformerer tilbake fra frekvsensdomenet.
2. Man kan se at det oppstår en tydelig svart ramme rundt hele det middelverdi-filtrerte bildet og en skurrete øverste i det frekvsensdomene-filtrerte bildet. Det skjer fordi origo til pikslene ved kanten vil hente verdier utenfra bildet, som er 0 og konvulsjonen da vil gi veldig lave verdier ved disse.

Oppgave 1.3

Grafen over viser grafen ut av program jeg lagde. Jeg eksperimenterte med filterstørrelser fra 0-20. Det ser ut til at det er optimalt å benytte frekvsensdomenet, foruten når man bruker et filter med størrelse 1-3. Man kan observere det som ser ut som eksponentiell vekst på kjøretid når man benytter romligdomenet. Filtrering i frekvsensdomenet er derfor det raskeste.

Oppgave 2

Originalbilde:



Kompresjon med $q = 0.1$:



Kompresjon med $q = 0.5$:



Kompresjon med $q = 2$:



Kompresjon med $q = 8$:



Kompresjon med $q = 32$:



Kompresjonsrate og lagringsplass spart:

```
Jobber...
Statistikk for kompresjon med q = 0.1: Kompresjonsrate = 2.0266655108891727, Lagringsplass spart= 50.65786659776613%
Statistikk for kompresjon med q = 0.5: Kompresjonsrate = 3.5462037837290055, Lagringsplass spart= 71.80083094524107%
Statistikk for kompresjon med q = 2: Kompresjonsrate = 6.858934787444641, Lagringsplass spart= 85.42047663391534%
Statistikk for kompresjon med q = 8: Kompresjonsrate = 15.24179548815422, Lagringsplass spart= 93.43909317785302%
Statistikk for kompresjon med q = 32: Kompresjonsrate = 41.54696196295287, Lagringsplass spart= 97.59308514328511%
Ferdig
```

Drøfting:

Jeg klarer å legge merke til rekonstruksjonsfeil når bildet er komprimert med $q = 2$. Dette uten zooming eller annen bildebehandlingssoftware. Spesielt himmelen blir mer kornete ved min umiddelbare gransking. Rekonstruksjonsfeilene blir så mer og mer tydelig jo høyere q går. Ved 0.5 ser jeg derimot ingen merkbare rekonstruksjonsfeil med mitt blotte øye. Dette vil jeg derfor si er "god nok". Som vi ser fra utskriften i terminalen tilsvarer komprimering med $q = 0.5$ en lagringsplassbesparelse på ca. 72%, noe som er veldig bra. Kompresjonsraten øker med hensyn på q fordi q tas in som parameter og multipleres med matrisen Q . qQ vil dermed vokse med hensyn på q og da får vi lavere tall ut av DC-transformasjonen. Lavere tall betyr at flere pikselverdier er like og vi vil dermed få høyere kompresjonsrate.