

## 0.1 Status omkring arbejdet

Gruppen starter med at fortælle vejleder hvad gruppen er i gang med at arbejde. Der fortælles at analysen er ved at blive færdiggjort, og at gruppen er ved at være klar til at begynde at programmere. Gruppen fortæller at der er lavet issues på GitHub så folk kan assignes. Algoritmerne er begyndt at være klar til brug i programmet. Det mangler nu at implementeres i kode. Vejleder foreslår at vi laver nogle tests af programmet hvis vi får tiden til det. Gruppen fortæller at vi laver et testprogram, der genererer en tilfældig graf. På den måde kan vi teste programmet på en hel masse grafer, hvor vi kan teste algoritmens korrekthed samt hastighed. Vejleder spørger om gruppen er blevet introduceret til unit tests, og foreslår at vi unit tester nogle af funktionerne. Dog behøver alle funktioner ikke at skulle unit testes.

## 0.2 Output af program

Vejleder spørger hvad gruppen har tænkt med hensyn til output af programmet. Gruppen fortæller at selve algoritmen outputter id'er af verticer. Det er så meningen at dataen kan repræsenteres af et andet program der bruger disse id'er til at hente nogle human-readable strenge.

## 0.3 Håndtering af flere etager

Vejlederne spørger hvordan vi har tænkt os at repræsentere flere etager i grafen. Gruppen fortæller at de har valgt at sige at en floor er unik. F.eks. kan etage 1 i bygning A kan have floor id 0, og etage 1 i bygning B kan have floor id 1. Derudover har gruppen tænkt at implementere note-types, som kan beskrive om en note er en trappe eller en elevator. Der forklares yderligere hvordan gruppen har tænkt sig at opfylde brugerens ønsker omkring ruten. Dette gøres ved at bruge en modifier, der eksempelvis giver en meget høj værdi af en f-værdi (A star) når der bruges en elevator. Der forklares ligelides med tegninger på tavlen hvordan gruppen har tænkt sig at bruge præcalculerede ruter, når der skal navigeres på tværs af etager. Disse præcalculerede ruter er statiske og kan derfor genbruges.

Gruppen har valgt at bruge en speciel metode til at kunne håndtere flere etager. Step 1: beregn afstand til alle output notes, som f.eks. trapper. Step 2. Brug præcalculerede ruter fra denne trappe til det ønskede måls etage. Step 3. Beregn afstand fra alle indgange på ønskede måls etage til selve målet. Alle resultater fra step 1, 2 og 3 lægges sammen. Den korteste rute vil så være den rute med laveste vægt.

## **0.4 Repræsentation af ruten**

Vejleder spørger hvordan vi har tænkt sig at repræsentere en rute. Dette kunne evt. beskrives i en perspektivering. Gruppen siger at en grøn pil kunne markerede ruten.

## **0.5 Billeder af tavlen**

Vejlederen foreslår at billeder af hvordan vi har tænkt os at lave grafen, skal laves pænt på computeren. Disse billeder kan bruges senere til at præsentere algoritmen.

## **0.6 Dokumentation af gruppens diskussioner**

Vejlederen banker i bordet og påpeger at det er VIGTIGT at dokumentere alle gruppens valg omkring algoritmen. Vejlederen foreslår at gruppen deler dokumentationen op i små letlæselige bidder.

Gruppen forklarer at de har haft diskussioner omkring hvorvidt algoritmen skal håndteres dynamisk eller statisk.

## **0.7 Omkring eksamen**

Vejlederen påminder om at huske de danske begreber på vertices, edges osv. Hvis fremlæggelsen er på dansk, skal graf teori begreberne også være på dansk. Der spørges fra gruppens side af, hvorvidt testgrafen også skal være på cd'en. Vejleder svarer at gruppen bare skal smide alt der er relevant for projektet, på cd'en.

## **0.8 Næste vejledermøder med vejleder**

Tirsdag d. 10. december kl. 8.30 og sidste vejledermøde: Mandag d. 16. december kl. 12.30