

Ruteplanlægning for turister



P1 PROJEKT
GRUPPE A401
SOFTWARE
AALBORG UNIVERSITET
DEN. 18. DECEMBER 2014



AALBORG UNIVERSITET
STUDENTERRAPPORT

**Første Studieår v/ Det Teknisk-
Naturvidenskabelige Fakultet**
Software
Strandvejen 12-14
9000 Aalborg

Titel:

Ruteplanlægning

Projekt:

P1-projekt

Projektperiode:

Oktober 2014 - December 2014

Projektgruppe:

A401

Deltagere:

Christian Dannesboe
Frederik Børsting Lund
Karrar Al-Sami
Mark Kloch Haurum
Mikael Sandegaard Aarsnes
Rabee Mohamad Kaddoura
Søren Lyng

Hovedvejleder:

Jane Billestrup

Bivejleder:

Mona-Lisa Dahms

Synopsis:

Projektet er udført ud fra projektoplægget "Bedre rutevejledning i Google Maps". Frem for en rute mellem ét punkt til ét andet punkt, ønskes en flerpunktsrute. Som case har gruppen overvejet forskellige grupper, hvor valget faldt på at fokusere på turister. Formålet med projektet er at lave en interessant flerpunktsrute, dog vil gruppen ikke diktere hvad den interessante rute er, derfor kan brugeren selv tilføje attraktioner, som vil udgøre personens interessante rute. Der er blevet undersøgt forskellige algoritmer, til at finde en relativ kort rute mellem attraktionerne, så ruten både vil være brugerens interessante rute, samt den forholdsvis korteste rute.

Oplagstal: 10

Sidetall: 70

Appendiks: 6

Afsluttet 18-12-2012

Rapportens indhold er frit tilgængeligt, men offentliggørelse (med kildeangivelse) må kun ske efter aftale med forfatterne.

Christian Dannesboe

Frederik Børsting Lund

Karrar Al-Sami

Mark Kloch Haurum

Mikael Sandegaard Aarsnes

Rabee Mohamad Kaddoura

Søren Lyng

Forord

Dette projekt er udarbejdet i et samarbejde mellem syv software-studerende på første semester fra Det Teknisk-Videnskabelige Fakultet på Aalborg Universitet.

I udarbejdelsen af projektet, har gruppen taget udgangspunkt i Aalborg-modellen, i form af problem- og projektbaseret læring. Der tages udgangspunkt i et problem, hvor læringen sker i form af projektarbejde i grupper.

Gruppen vil gerne sige tak til hovedvejlederen Jane Billestrup og bivejleder Mona-Lisa Dahms, for deres vejledning gennem projektet. Derudover vil gruppen også gerne takke Lars Bech og Kim Mikael Jensen fra VisitAalborg, for at stille op til et interview med gruppen.

Læsevejledning

Kildehenvisning

I dette projekt bruges Harvard-metoden, også kendt som Chicago-metoden, til kilde henvisning. Hvis der henvises til en bestemt kilde, efter eksempelvis en sætning, påstand eller et citat, henvises der på følgende måde: Sætning/påstand/citat [Forfatter, udgivelsesår].

Hvis kilden anvendes til hele afsnit, sættes kildehenvisningen efter punktummet, således: Afsnit.[Forfatter, udgivelsesår]

I afsnittet "Litteratur" vil kilde henvisningerne blive sorteret i alfabetisk rækkefølge. Hvis det eksempelvis var en hjemmeside der blev brugt som kilde, ville det se således ud:

Kilde henvisningen fra rapporten. Forfatter. *Titel*. URL. Udgivelsesår. Dato siden er set og evt. sidetal.

Dansk Statistik, 2008. Dansk Statistik. *Turismen – Regionalt, nationalt og internationalt*. <http://www.dst.dk/pukora/epub/upload/11676/tur08.pdf>, 2008. Set d. 19/11-2014 – side 8.

Hvis nogle af disse informationer mangler, eksempelvis udgivelsesår, udelades de.

Figurhenvisning

Igennem rapporten vil der blive henvist til figurer og illustrationer, hvor det vil blive anvist ud fra hvilket afsnit det befinder sig i, samt hvilket nummer figuren er i det omtalte afsnit. Herudover skal der være en beskrivende tekst, der forklarer figuren, eksempelvis:

Figur 2, afsnit 5: Figurtekst

Figur 5.2: Figurbeskrivelse

Indholdsfortegnelse

Kapitel 1 Indledning	1
1.1 Begreber	2
Kapitel 2 Problemanalyse	3
2.1 Interessentanalyse	3
2.1.1 Prioriteringen	4
2.1.2 Opsummering	5
2.2 Spørgeskema	5
2.2.1 Udformning	5
2.2.2 Resultatbehandling	5
2.2.3 Opsummering	7
2.3 Interviewet	7
2.3.1 Udformning	7
2.3.2 Resultatbehandling	7
2.3.3 Opsummering	8
2.4 Eksisterende løsninger	8
2.4.1 TripAdvisor Offline City Guides	8
2.4.2 FindTheBestRoute.com	9
2.4.3 Opsummering	9
Kapitel 3 Problembeskrivelse	10
3.1 Problemafgrensning	10
3.2 Problemformulering	10
Kapitel 4 Kravspecifikationer	11
4.1 Optimale løsningsforslag	11
4.2 Gruppens løsningsforslag	13
Kapitel 5 Teori	15
5.1 Grafteori	15
5.1.1 Begrebsbeskrivelse	15
5.1.2 Nearest neighbour algoritmen	16
5.1.3 Dijkstra's algoritme	16
5.1.4 Opsummering	17
5.2 Vektorteori	17
5.3 Positionsbestemmelse	20
Kapitel 6 Implementering	21
6.1 Programbeskrivelse	21
6.2 Beskrivelse af structs	21
6.3 Beskrivelse af funktioner	22

Kapitel 7 Test	31
7.1 Testcases	31
Kapitel 8 Diskussion	34
8.1 Spørgeskema	34
8.2 Interview	35
8.3 Programmet	35
8.4 Den interessante rute	36
Kapitel 9 Konklusion & Perspektivering	37
Litteratur	39
Appendiks A Spørgeskema	41
A.1 Teori	41
A.1.1 Spørgsmålene	41
A.1.2 Formulering	41
A.1.3 Rækkefølge	42
A.2 Rådata	42
Appendiks B Interview	45
B.1 Teori	45
B.1.1 Enkeltinterview	45
B.1.2 Telefoninterview	45
B.1.3 Spørgeteknikker og metoder til interview	46
B.1.4 Lukkede og åbne spørgsmål	46
B.1.5 Passive teknikker	46
B.1.6 Aktiv spørgeteknik	46
Appendiks C Interviewguide	47
Appendiks D Transskribering	48
Appendiks E Program koden	55
Appendiks F Nyt p1 forslag	63

Hvert år besøger flere millioner turister Danmark, hvilket er godt for den danske økonomi. Når turisterne bruger penge på en dansk vare, service eller oplevelse, bliver det sådan set eksporteret til udlandet – og derfor bliver dette betragtet som en eksportvare. I alt står denne eksport type for 3,6% af den danske eksport. Turisterne har et forbrug på 87,2 mia. kr., hvoraf de udenlandske turister bruger 35,7 mia. kr. altså godt 41%, mens de danske turister står for de resterende 59%. Udover at turismen hjælper det danske samfund økonomisk, skaber turismen ifølge VisitDanmark knap 122.500 fuldtidsjobs. [VisitDanmark, 2013]

Det ses gerne at turisterne kommer tilbage til Danmark igen. Dette sker naturligvis ved, at turisterne nyder deres ophold og får den bedst mulige ferie. Som turist i en storby kan det forekomme svært at finde rundt, og samtidigt virke let at fare vildt. Hvis en turist i København gerne vil se Rundetårn, kan turisten kigge efter det store monument, og gå i den retning hvor attraktionen nu er. Dog kan det ske at turisten undervejs mister tårnet af syne, og pludseligt ved turisten ikke i hvilken retning personen nu skal gå. Turisten kan vælge at bruge sin smartphone, hvis turisten da er i besiddelse af en, og kan eksempelvis gå på internetsiden GoogleMaps. Dette har de danske turister mulighed for, men for en udenlandsk turist vil dataen de bruger på ferie, fra deres smartphone koste mere, end det ville hjemme i deres hjemland [Vodafone, 2014]. Her kan turisten så finde en rutevejledning fra punkt A til B, dog vil der kunne opleves problematikker, hvis en flerpunktsrute ønskes. I dette projekt har gruppen valgt at afgrænse sig til at undersøge en enkelt dansk by, i dette tilfælde valgte gruppen Aalborg, da dette var mest oplagt.

For turisten vil planlægning på forhånd være en god ting, hvis turisterne vil nå så mange attraktioner som muligt på en ferie, da tiden kan være begrænset [Youth Central, 2014]. Hvis feriedestinationen er i udlandet, ville en offline løsning til ruteplanlægning være optimal, da brugen af mobildata i udlandet kan koste mange penge [TDC, 2014].

Samtidigt kan der spørges, hvad der gør en rute god: Er det hvor hurtigt turisten kommer fra den ene valgte attraktion til den anden? Kan der findes en mere interessant rute, med attraktioner, som turisten ikke har kendskab til? Der kan være mange parametre der spiller ind, når den foretrukne rute skal vælges.

I dette projekt har gruppen valgt at undersøge:

I hvilket omfang kan ruteplanlægning hjælpe turister med, at finde den korteste eller mest interessante rute mellem attraktioner i Aalborg?

Herunder kan der undersøges hvad en interessant rute ville være, hvilken form for rute turisterne foretrækker og i hvilke situationer turister har brug for ruteplanlægning og hvorfor? Derudover vil gruppen undersøge hvilke typer af turister der er relevante i forhold til dette projekt og hvad deres behov kunne være.

1.1 Begreber

Ved en 'interessant rute' vil der i denne rapport refereres til en rute, der indkluderer unikke kulturelle og nationale oplevelser, hvor det interessante aspekt er individuelt for hver turist. Steder der ville kunne gøre ruten mere interessant vil bl.a. være havnefronten, gågaden eller andre attraktioner turisten ikke selv havde tænkt på, som turisten ville kunne passere på sin vej fra A til B.

En 'attraktion' vil i denne rapport forstås som både traditionelle, og utraditionelle attraktioner. Traditionelle attraktioner i Aalborg er Aalborg Zoo, kunstmuseet og lignende. De utraditionelle indkluderer steder som Storcenteret, Ikea og Havnefronten.

'Turist' vil i denne rapport være til dels defineret ud fra UNWTOs (UN World Tourism Organisation) beskrivelse, som beskriver to slags turister. En-dagsturister, som maksimalt overnatter én nat på stedet, og en generel turist, som overnatter mere end en nat [Danmarks Statistik, 2008]. Udover disse to typer findes undergruppen 'erhvervsturister'. Dette er turister, der kommer til stedet med et arbejdsformål.

En turist er en person, der har et bestemt formål med rejsen, til forskel fra besøgende og rejsende. Formålet kan fx være, at opleve den danske kultur.

En 'storby' i Danmark, er i denne rapport en by, omgivet af områder med lavere bebyggelse. Hvor Aalborg fx har Nørresundby og Vejgaard.

Med dette afsnit har gruppen dannet grundlag for analysen, hvori de beskrevne begreber vil blive brugt. I de kommende afsnit vil den initierende problemstilling og dets spørgsmål blive analyseret.

Problemanalyse 2

I dette kapitel vil den initierende problemstilling blive analyseret. Dette bliver gjort i form af en interessentanalyse, hvor de væsentlige interessenter bliver fundet. På baggrund af interessentanalysen blev der sendt et spørgeskema ud, og et interview blev foretaget med VisitAalborg. Til sidst i dette kapitel er to eksisterende løsninger blevet analyseret.

2.1 Interessentanalyse

Gruppen vil i dette afsnit undersøge diverse personer/grupper, der kan fungere som interessenter i projektet, altså en person der vil have nytte af eller kan bidrage til projektet. Herefter vil gruppen prioritere disse interessenter, alt efter hvor relevante de er i forhold til projektet.

Turister

Turister er en væsentlig interessant i projektet, da en turist ofte vil se hvad byen har at byde på, eller nogle unikke attraktioner, i forhold til følgende citat:

“If you want visitors to come back again — and say nice things about your town to others who might come, too — you need to have some good answers at the ready. That means offering things to see and do that are either unique or extraordinary...” [Vitcenda, 2013].

Hvis turisterne planlægger hvad det er de vil se, kan turisterne spare tid [Youth Central, 2014]. Turisterne kan vælge at gå en længere rute, og derved have mulighed for at finde andre ting, som de vælger at bruge deres tid på. Et ruteplanlægningsværktøj vil derfor være interessant for turister, da de derved kan komme til at besøge alle de attraktioner/seværdigheder, som de ønsker. En-dagsturister er en mindre interessant i projektet, da en en-dagsturist maksimum overnatter en nat, og generelt har et planlagt formål med rejsen, som fx at besøge Storcenteret, venner/familie eller er på et kursusophold [Center for Erhverv & Udvikling, 2012].

Erhvervs- og Vækstministeriet

Erhvervs- og Vækstministeriet (EVM) er en interessant i projektet, da det er dette ministerie, turisme indgår under, da turismen er et erhverv der beskæftiger mange tusinde mennesker. Hvis der er nogle unikke eller ekstraordinære attraktioner i en by, vil turister huske disse, som gode oplevelser, og nogle vil derfor komme igen. Det er noget som EVM er interesseret i, da der kommer flere penge ind i landet [Erhvervs- og Vækstministeriet, 2014]. Her vil et ruteplanlægningsværktøj kunne hjælpe turister med at se nogle attraktioner, hvis der fx er en top 5 over de attraktionerne der er i landet, eller i den by ferien foregår.

Retail-handel

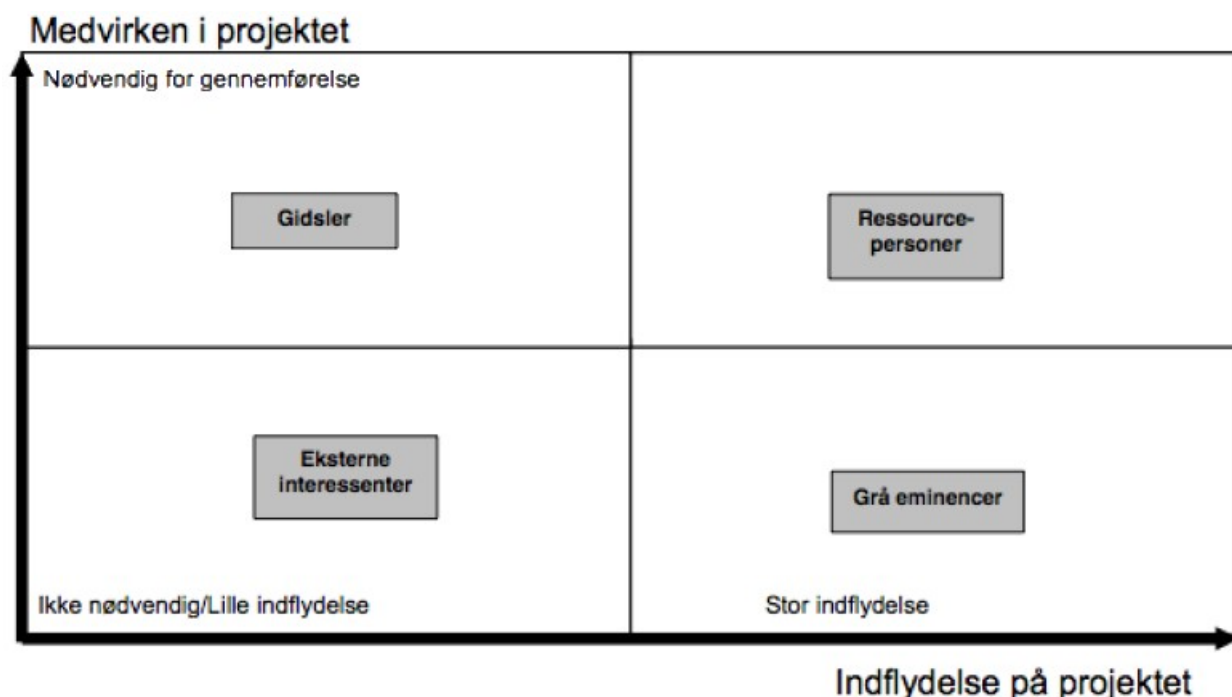
Diverse forretninger er også interessenter i projektet, da kendte brands som fx IKEA, Bilka og lignende er så store, at de kan tiltrække turister [Ritzau, 2014]. Ved at implementere disse adresser i et ruteplanlægningsværktøj, kan det tiltrække turister, og derved øge omsætningen, til gavn for forretningerne.

VisitAalborg

VisitAalborgs arbejde består af at støtte aktører, hvis disse aktører byder på nogle turistfremmende aktiviteter/projekter, der har til formål at hjælpe Aalborg. Det kan fx være at henvise til aktørernes hjemmesider, gennem deres egen hjemmeside, der bliver set af ca. 600.000 årligt. [VisitAalborg, 2014] Til et ruteplanlægningsværktøj er VisitAalborg en vigtig interessant, da de har informationer om turisterne i Aalborg. Ved at inddrage turistkontoret i projektet, vil gruppen gøre brug af deres ressourcer.

2.1.1 Prioriteringen

For at prioritere interessenterne i projektet, og finde de vigtigste interessenter, har gruppen valgt at gøre brug af indflydelse/medvirken-matrixen, som kan ses på figur 2.1 herunder.



Figur 2.1: Indflydelse/medvirken-matrixen

Indflydelse/medvirken-matrixen opdeler interessenterne i fire grupper, og giver gruppen et overblik over, hvilke interessenter der skal tages højde for, og hvem der kan bidrage med vigtig information og ressourcer til projektet.

VisitAalborg er i dette projekt ressourceperson, da de har informationer om turisme. Derudover kan turistkontoret komme med råd og vejledning for, at en eventuel løsning vil være mest optimal for turisterne.

Turisterne er også kategoriseret som ressourceperson, da turisternes viden og erfaringer fra deres storbyferier, kan bruges til projektet.

De eksterne i projektet er retail-forretningerne og Erhvervs- og Vækstministeriet. Deres indflydelse og medvirken er ikke nødvendig, for at kunne gennemføre projektet, men projektet kan stadig have en indflydelse på dem.

Gruppen har i dette projekt ikke nogle interessenter, der passer ind under kategorierne grå eminence og gidsler.

2.1.2 Opsummering

Ud fra interessentanalysen, er gruppen nået frem til, at VisitAalborg og turisterne er ressourcepersoner, hvilket gruppen ser som en centrale interessenter i projektet. Det er nødvendige kilder, for at få nogle brugbare ressourcer. De andre interessenter, som gruppen har fundet frem til, er blevet vurderet som mindre vigtige, og derfor vil fokuset ligge hos VisitAalborg og turisterne. Da turister og turistbureauet er projektets væsentligste interessenter, har gruppen valgt at uddrage information fra disse to interessenter i form af et spørgeskema og et interview. I tilfælde af VisitAalborg og turisternes interesser modstrides, vil gruppen vægte turisternes interesser højest, da VisitAalborg kan have økonomiske interesser i at fremme bestemte partnere, fx Aalborg Zoo, Nordkraft og Aalborg Kongress og Kulturcenter [VisitAalborg, 2014].

Det kommende afsnit vil omhandle informationsøgning, i form af det udsendte spørgeskema.

2.2 Spørgeskema

I dette afsnit vil udformningen af spørgeskemaet og de resultater, der kan udtrækkes fra spørgeskemaet, blive belyst. I interessentanalysen blev turister placeret som ressourceperson, og ud fra dette har gruppen valgt at lave et spørgeskema, som skal hjælpe med at besvare den initierende problemstilling. Dette blev gjort med den hensigt at få så mange synspunkter på de opstillede problemstillinger, så det ville være muligt at generalisere ud fra besvarelserne. Spørgeskemaet blev udarbejdet vha. den kvantitative metode. Se i appendix A1 og A2 om mere information om den brugte teori og rådata.

2.2.1 Udformning

Først udformede gruppen en række problemstillinger, som gruppen ville have svar på i undersøgelsen:

- Hvilke slags attraktioner tager turister til storbyerne for at se?
- Hvordan planlægger turister deres ferie?
- Hvordan finder turister rundt?
- Har turister problemer med at finde rundt?
- Fortrækker turister den hurtigste eller den mest interessante rute?
- Er den løsning gruppen har i tankerne noget respondenterne ville bruge?

Gruppen valgte at dele spørgeskemaet på Facebook. Dette vil dog give nogle begrænsninger: Spørgeskemaet kan kun ses af personer som gruppen er venner med på Facebook, hvilket mest vil være unge mennesker. Derudover er spørgeskemaet lavet på dansk, for at gøre spørgsmålene så let forståelige som muligt for gruppens venner på Facebook. Da mange af de personer gruppen kender på Facebook sikkert ikke har været turister i Aalborg, blev spørgsmålene rettet mod alle storbyer.

2.2.2 Resultatbehandling

Spørgeskemaet var lagt op på Facebook i tre dage, hvorefter resultaterne blev behandlet. I alt var der kommet 60 besvarelser, hvilket giver gruppen en relativ lille respondentgruppe at arbejde med, dog giver resultaterne nogle klare tendenser, som gruppen har arbejdet ud fra. Herunder kan der ses forklaring på formålet med spørgsmålet, og hvad besvarelserne kan fortælle.

Hvad er vigtigt for dig på din storbyferie?

Formålet med dette spørgsmål er at finde ud af hvad turister gerne vil se eller opleve i en storby, for at kunne se hvad der eventuelt kunne implementeres i projektets løsning.

Ud fra besvarelserne, afgivet af respondent gruppen, kan der ses hvad respondenterne vægter højest på deres storbyferie. De mest populære oplevelser var i følge spørgeskemaet: Se byens seværdigheder 90%, opleve kulturen 68.33%, maden 63.33% og shopping 50%. Derudover var der en del der havde kommenteret, at de kom til storbyen for at se sportsbegivenheder. Dette var dog ikke en valgmulighed på spørgeskemaet, så det kan ikke uddrages til projektet, da der kan have været en mulighed for at respondenterne ikke havde tænkt over dette svar.

Hvilke hjælpemidler bruger du til at planlægge din storbysferie?

Formålet med dette spørgsmål er, at finde eksisterende planlægningsværktøjer, hvor fordele og ulemper til dette projekt kan undersøges.

I besvarelserne kan der ses en liste (Appendix A2) af eksisterende planlægningsværktøjer, hvor de både er elektroniske og i form af brochurer og lignende.

Hvilke redskaber bruger du til at finde rundt når du er på storbyferie?

Formålet med dette spørgsmål er at finde ud af om turister brugte redskaber til at finde rundt. Ud fra besvarelserne på dette spørgsmål kan der ses, hvilke redskaber respondentgruppen bruger, til at finde rundt på ferier. Her har respondentgruppen mulighed for at svarer på mere end én ting. Der kan ses at 81,67% bruger diverse kort og brochurer, og 55% bruger elektroniske redskaber til at finde rundt.

Har du nogensinde haft problemer med at finde vej på din storbyferie?

Formålet med dette spørgsmål er, at få afklaret om det er et problem for turisterne at finde rundt på deres storbyferier.

Ud fra besvarelserne kan der ses, at hele 68,33% altså hele 2/3, har haft problemer med at finde rundt på deres storbyferie.

Når du skal fra en aktivitet til en anden på din storbyferie, vil du helst tage den hurtigste rute eller en langsommere men mere interessant rute?

Formålet med dette spørgsmål er, at finde ud af hvad turisterne helst ville have, en hurtig rute fra A til B, eller en interessant rute, som er langsommere, hvor turisten får set andre ting på vejen til sin destination.

Der ses tydeligt at respondenterne foretrækker den interessante rute over den hurtigste rute, med henholdsvis 80% for den interessante og 20% for den hurtigste rute.

Et program/applikation, som hjælper mig med at finde den hurtigste og/eller mest interessante vej igennem byen, via brugerens valgte "must see" destinationer, ville være noget jeg kunne bruge?

Formålet med dette spørgsmål er at finde ud af om projektet egentlig har nogen interesse hos brugeren.

I dette spørgsmål svarede 90% at den ideelle løsning ville enten kunne bruges eller ønskes.

2.2.3 Opsummering

Ud fra besvarelsenerne fra spørgeskemaet, blev der konstateret at det er et problem, at turisterne farer vildt på deres storbyferie, da 68,33% af respondenterne har haft problemer med at finde rundt på deres storbyferie, samt hele 80% gerne ville have en interessant rute, frem for den hurtigste rute. Ved at gå den interessante rute, får turisterne set mere af byen de befinder sig i, dette kunne eksempelvis være havnefronter og diverse statuer. Det viste sig også at mange turister stadig bruger kort og brochurer, når de skal finde rundt på deres storbyferie, hvilket var hele 81,67%, hvor der i alt er 55%, som bruger elektroniske redskaber til finde rundt, dog skal det pointeres at respondenterne havde mulighed for at krydse mere end en besvarelse af. Derudover mente lige godt 90% at et program/applikation, som hjælper turisten med at finde den hurtigste og/eller mest interessante vej gennem byen, via brugerens valgte "must see" destinationer, ville kunne bruges.

Det kommende afsnit omhandler interviewet foretaget med Lars Bech og Kim Mikael Jensen, fra VisitAalborg.

2.3 Interviewet

På baggrund af interessentanalysen kontaktede gruppen VisitAalborg, for at lave et interview vedrørende turisme i Aalborg. Dette interview blev forberedt som et semistruktureret interview, med en passiv spørgeteknik. Formålet med interviewet var, at fremskaffe viden og information fra en professionel kilde, der selv arbejder med emnet.

2.3.1 Udformning

Gruppen valgte at interviewet skulle være et semistruktureret enkeltinterview med passiv spørgeteknik. Herved kan respondenterne snakke frit, så der er mulighed for længere uddybende svar, som ikke var tiltænkt af gruppen på forhånd. Disse teknikker blev brugt, da gruppen ikke havde meget viden om emnet på daværende tidspunkt, og derfor gav dette mulighed for mere information, end der var tiltænkt. For mere information om spørgeteknikkerne, se appendix B1.

Gruppen udformede en interview-guide, som kunne bruges under interviewet. Da respondenterne var meget snaksagelig, blev denne guide ikke brugt til fulde. Ikke alle forberedte spørgsmål blev stillet, men blev dog besvaret løbende i interviewet. Spørgsmålene anvendt i interview-guiden, blev udformet efter gruppens initierende problemstilling og tilhørende underspørgsmål.

2.3.2 Resultatbehandling

Lars Bech nævner i interviewet, at den mest anvendte metode til bestemmelse af antal turister i Aalborg, er gennem målinger foretaget af Horesta, som gøres tilgængelig gennem Danmarks Statistik. Dette kan VisitAalborg bruge til at se fremgang i turismen i Aalborg. Hvis der ønskes resultater til antallet af turister, der overnatter i Aalborg, kan Danmarks Statistik benyttes. Dog gør Lars opmærksom på, at der også findes turister, som ikke overnatter, hvilket gør det svært, at finde det reelle antal turister, der besøger Aalborg. VisitAalborgs medieudvikler Kim Mikael Jensen oplyser, at en udvidet løsning lignende TripAdvisor ville være interessant for både turisten og VisitAalborg, samtidigt nævner han, at de ikke kender til en lignende løsning endnu. Udfra dette kan det bekræftes, at der er interesse for en løsning, dog gør Kim opmærksom på, at

tidligere forsøg på løsninger, har været for komplicerede og har for mange funktioner. Han søger simplicitet i et program.

Under interviewet diskuterer Lars, hvilke attraktioner der er interessante, ifølge tal, og kommer også ind på, hvad han selv mener er attraktioner i Aalborg. Som eksempler nævner han blandt andet IKEA, Aalborg Zoo og kunstmuseet, hvor han selv også mener, at gågaden, havnefronten og danske butikker (Georg Jensen, Inspiration osv) også er attraktioner. ”Hvis man køber et eller andet, som man er glad for, så kan man altid huske hvor man har købt det henne.” Dette citat, taget fra interviewet med Lars, passer overens med citatet fra interessentanalysen.

2.3.3 Opsummering

Igennem interviewet blev dele af problemstillingen bekræftet i form af, at et eventuelt program er ønsket. Der blev desuden foreslået krav til en eventuel løsning. Kim Mikael Jensen sagde, at programmet skal være simpelt, den skal altså ikke have for mange funktioner. Lars gjorde det også mere klart, hvilke attraktioner i Aalborg, der er interessante, fx IKEA, Aalborg Zoo og kunstmuseet. Interviewet er også blevet brugt til bestemmelse af begrebet ”attraktion”.

I det efterfølgende afsnit vil to eksisterende løsninger blive analyseret. De eksisterende løsninger der vil blive analyseret er henholdsvis TripAdvisor og findthebestroute.com.

2.4 Eksisterende løsninger

I spørgeskemaet blev der opremset en række af hjælpemidler som respondenterne bruger på deres storbyferie, hvilket til projektet vil være eksisterende løsninger. I det kommende afsnit vil der blive analyseret på TripAdvisors app, hvilken var en af de hjælpemidler som respondent gruppen havde nævnt. Der udover har gruppen fundet Findthebestroute.com, som vha. Google Maps kan lave en flerpunktsrute.

2.4.1 TripAdvisor Offline City Guides

TripAdvisor har en offline app, der kan hjælpe med at guide turister rundt, i den by de er rejst til. Den har mange forskellige funktioner, som fx et kort indlagt i appen. Dette kort kan være effektivt, hvis brugeren har forberedt sig hjemmefra. Dette skyldes, at turisten kan downloade et kort over den by, brugeren skal besøge, og derved vil den fungere offline. Grunden til at det er effektivt for turister, skyldes at mobildata kan være dyrt i udlandet[TDC, 2014].

Udover et kort, har app'en også nogle informationer omkring de mange forskellige byer. Disse informationer har turisten ligeledes mulighed for at downloade, så de også er tilgængelige offline. Ved hjælp af disse informationer, kan turisten fremskaffe sig hjælp, hvis turisten fx er interesseret i at finde en restaurant, et hotel, se en bestemt attraktion eller lignende. Ønsker brugeren fx at besøge en attraktion, kan der ved hjælp af en knap, klikkes frem til en lokation, som turisten enten selv kan finde vej til, eller benytte en anden knap i app'en og få indlagt ruten i det downloadede kort.

Hver af disse kategorier indeholder en ”Best in Town”-funktion, som er en liste over de mest populære attraktioner, ifølge TripAdvisors brugere, da der er et point-system, som giver brugere af app'en mulighed for at vurdere og skrive kommentar til de enkelte attraktioner, på en skala fra 1-5.

TripAdvisors app har mange gode funktioner. En af de gode funktioner, er det offline kort, der giver mulighed for at undgå brugen af mobildata på en udlandsrejse, og gør det muligt hele tiden

at have et kort ved hånden. Herudover kan der fås et indblik i, hvilke ting der er at se og opleve i den valgte by, med kommentarer og ratings fra andre brugere, der har besøgt disse steder. Appen har også nogle mangler, som fx at vælge flere seværdigheder på listen, og give en rute mellem disse seværdigheder, så det er muligt at få en flerpunktsrute.

2.4.2 FindTheBestRoute.com

Google Maps er begrænset til kun at kunne vise vejen fra et punkt til et andet. Det har FindTheBestRoute.com taget kampen op imod, og har derfor lavet en hjemmeside på FindTheBestRoute.com, hvor den hurtigste rute mellem maksimalt 10 forskellige adresser kan beregnes. FindTheBestRoute.com, udnytter Google Maps JavaScript API v3, altså en grænseflade til Google Maps, der tillader andre programmer at benytte Google Maps, til fx at få vist et kort, eller beregne en rute [FindTheBestRoute.com, 2014].

Selvom der på maps.google.dk ikke er mulighed for at indtaste forskellige destinationer, og få anvist den hurtigste rute imellem punkterne, så har Google Maps allerede funktionaliteten indbygget til at foretage denne beregning, som en løsning på "The Travelling Salesman Problem". For findthebestroute.com, er det derfor simpelt at sende en anmodning til Google, der indeholder informationer om de forskellige destinationer der skal forbindes med en rute. Google foretager så beregningerne, og sender den bedste rute tilbage til findthebestroute.com, hvor de så kan vise ruten til deres brugere [Google, 2014].

2.4.3 Opsummering

TripAdvisor har mange gode funktioner, så som offline kort, "Best in Town" og information om de enkelte attraktioner. TripAdvisor findes som en app, så den er tilgængelig på ferien. TripAdvisor har dog ikke mulighed for at lave en flerpunktsrute.

FindTheBestRoute.com har intet af det som Tripadvisor har, den har dog muligheden for at lave en flerpunktsrute, med maksimalt 10 adresser. Problemet med FindTheBestRoute.com er at den kun er tilgængelig på internettet og derved er den mere besværlig på ferien.

Disse synspunkter kan bruges til at udforme krav til projektets løsning.

Næste afsnit vil omhandle problembeskrivelsen, som indeholder problemafgrænsningen og den endelige problemformulering.

Problembeskrivelse 3

I dette afsnit vil den endelige problemformulering blive beskrevet.

3.1 Problemafgrænsning

Ud fra denne analyse, kan der konkluderes, at VisitAalborg og turisterne er ressourcepersoner, hvoraf turisterne har større indflydelse på programmet, da de er de endelige brugere. VisitAalborg kan bruges som guider til, hvad der kan være af indhold i programmet, men der skal stadig tages højde for, at det er i deres interesse, at få deres arbejds partnere med ind i programmet, selvom det ikke i alle tilfælde er til turistens interesse. Ved spørgeskemaet blev der uddraget, at turister allerede har løsninger fra punkt til punkt, hvor der i eksisterende løsninger blev påpeget, at der også findes løsninger for flerpunktsruter. TripAdvisor viste, at der også er programmer, som tager højde for brugerens interesse, men alt taget i betragtning, er der ikke en løsning der kombinerer alle disse funktioner, som der belyses i gruppens spørgeskema og interview, til at være i brugerens bedste interesse: En simpel løsning, der inddrager brugerens interesse, og foreslår yderligere punkter til en mere interessant rute. Dette kunne endda optimeres ved en offline-funktion, som TripAdvisor også gør brug af ved et offline kort.

3.2 Problemformulering

Hvordan udvikles der en softwareløsning, der hjælper turisten med at finde rundt i en storby, på en interessant rute mellem turistens egne valgte attraktioner?

- Hvordan kan programmet finde en rute?
- Hvordan kan programmet gøre ruten interessant?
- Hvordan kan programmet hjælpe brugeren med at finde rundt?

Kravspecifikationer 4

I dette afsnit vil gruppen vurdere, hvilke krav der skal indgå i en løsning. Heri vil der både blive opstillet krav til en optimal løsning og til en afgrænset løsning, som gruppen mener vil være realistisk, at kunne lave.

4.1 Optimale løsningsforslag

For at kunne udvikle en softwareløsning, der besvarer gruppens problemformulering, er det vigtigt at definere nogle krav til programmet. Til en optimal løsning, har gruppen vurderet, at der skal være følgende krav:

- Programmet skal udvikles som en applikation til moderne smartphones, inklusiv iOS, Android og Windows Phone.
- Programmet skal hjælpe brugeren, til at lave en interessant rute gennem byen, vedkommende vil besøge.
- Programmet skal vise et overskueligt og scalerbart kort med ruten.
- Ruten skal udskrives i samme stil som Google Maps.
- Programmet skal kunne give rutevejledning undervejs på ruten.
- Programmet skal kunne downloade en offline version af den rute der er valgt, inklusiv kortet for det omkringliggende område.

Til disse krav har gruppen konstrueret nogle skitser og en beskrivelse af den optimale løsning, for at give et billede af hvordan det eventuelt kunne se ud.

Gruppen har valgt at lave den ideelle løsning på følgende måde:

Brugeren bliver præsenteret for en liste over alle attraktioner i byen, sorteret efter afstand fra brugerens nuværende position, alfabetisk rækkefølge eller rating.

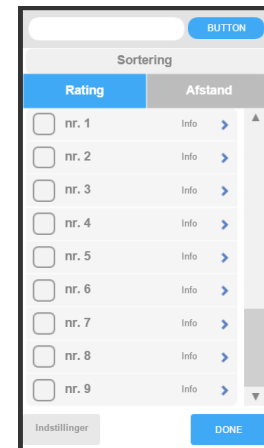
Ratingen vil blive fundet ved at tage gennemsnittet af alle brugeres rating af den specifikke attraktion. Dette gøres for at give brugeren mulighed for at vælge de attraktioner vedkommende gerne vil se.

Programmet skal under udvælgelsen give brugeren mulighed for at læse om de enkelte attraktioner, så brugeren kan foretage informerede beslutninger om til- og fravælgelse af attraktioner.

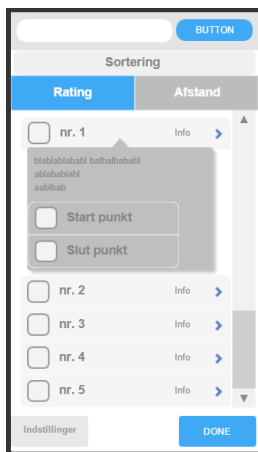
Når brugeren har valgt de attraktioner vedkommende vil besøge, udformes et kort med den korteste rute mellem disse attraktioner. På dette kort vil alle attraktioner som ligger indenfor en forudbestemt afstand til ruten, blive vist. Derefter kan brugeren tilføje nogle ekstra attraktioner, som vil blive tilføjet til den endelige rute.

Dette gøres så brugeren kan udforme en mere interessant rute. Gruppen vil ikke diktere hvad den interessante rute er, men lade brugeren selv tilvælge, og derved få deres egen unikke interessante rute.

Den optimale løsning vil som startside have en liste over alle attraktionerne. Disse skal kunne sorteres, til det findes to valg muligheder, forholdsvis rating og afstand, hvoraf rating viser en attraktionerne i rækkefølgen, baseret på hvad tidligere brugere har valgt at rate den. Funktionen afstand, vil vise hvor stor en afstand der er fra det punkt hvor brugeren står, til en attraktion. Attraktionerne vises da således, at den nærmeste attraktion er øverst. De attraktioner, som brugeren ønsker at se, skal brugeren blot trykke/klikke på og et flueben vil så komme frem i boksen til venstre. Efter alle de ønskede attraktioner er valgt, trykkes på done for at komme videre. Figur 4.1 viser en skitse af brugergrænsefladen.



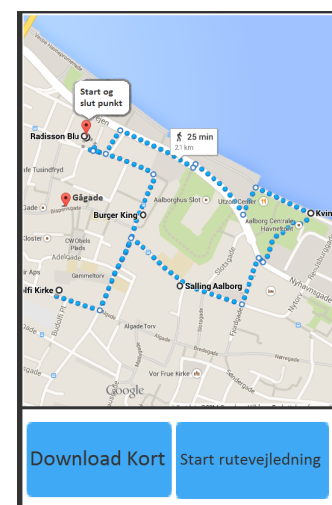
Figur 4.1:
Brugergrænseflade



Figur 4.2:
Eksempel på dropdown boks

Hvis brugeren ønsker mere information om en bestemt attraktion, vil vedkommende have mulighed for at trykke på "info" eller den lille pil i højre side. Dette vil få en dropdown boks til at komme frem med information om attraktionen. Derudover vil der i denne boks, være mulighed for at vælge et start- eller slutpunkt. Disse punkter skal give brugeren mulighed for at vælge, hvor brugeren ønsker at starte/-slutte sin rute. Figur 4.2 viser en skitse af denne dropdown boks.

Når der er valgt nogle ønskede destinationer/attraktioner, og trykket done på forrige side, skal programmet fremvise et kort med en rute mellem de valgte punkter. Ruten skal vise hvor lang hele ruten er, og hvor lang tid det tager at gå ruten. På kortet skal der ydermere vises de attraktioner som ligger indenfor en hvis afstand til ruten. Disse vises som punkter med et navn tilknyttet. Der skal desuden være to funktioner, når ruten bliver vist. Der skal være mulighed for at downloade kortet på mobilen, og derved gør det muligt at anvende programmet, uden brug af internet. Den anden funktion skal starte rutevejledningen, som fungerer som en ganske almindelig GPS-rutevejledning. En skitse af dette kan ses på figur 4.3.



Figur 3.3: Rute - Ruten er
hentet fra Maps.Google.com



Figur 4.4: Udvidet rute -
Ruten er hentet fra
Maps.Google.com

Når brugeren trykker på de attraktioner som er med på kortet, men ikke med i ruten, dette er, som beskrevet ovenfor, de attraktioner som ligger indenfor en hvis afstand til ruten, vil en boks komme frem med informationen om den givne attraktion. Der vil også være mulighed for her at tilføje denne attraktion til ruten. Figur 4.4 viser hvordan det eventuelt kunne laves.

4.2 Gruppens løsningsforslag

Gruppen har gennem spørgeskema og interview fået stillet en række krav til løsningen, af respondentgruppen og VisitAalborg. Gennem spørgeskemaet, blev det konkluderet, at det vigtigste for turister, er at de kan opleve byen på en interessant rute. Derudover har turistbureauet givet udtryk for, at løsningen gerne skal være så enkelt som muligt, altså færrest mulige funktioner, så brugeren ikke bliver forvirret, da de mener, at det er i turistens bedste interesse. Der er blevet stillet krav fra universitets side, om at programmet skal være et lille specifikt program i C, af høj kvalitet. Dette stemmer godt overens, med de krav der er blevet stillet fra turistbureauets side.

Ud fra dette, har gruppen opsat nogle krav for gruppens løsningsforslag, og de er som følgende:

- Programmet skal kunne beregne en kort rute mellem en række punkter.
- Programmet skal beregne om der ligger en attraktion mindre end 100 meter fra ruten, og tilføje ekstra attraktionen til ruten, hvis brugeren ønsker det.
- Programmet skal som output give en liste over rutens destinationer, i den rækkefølge de skal besøges, i forhold til den korte rute.

Da dette er et P1 projekt, og gruppen er begrænset af både tid og erfaring, har gruppen valgt at begrænse softwareløsningen, på følgende punkter:

- Afstanden mellem attraktioner vil blive beregnet i fugleflugtslinje.
- Brugeren kan kun vælge destinationer ud fra en række forudbestemte punkter.
- Tekstbaseret brugergrænseflade.

På baggrund af kravene og afgrænsningen, har gruppen valgt at lave et program, som har nogle forudbestemte destinationer, der dækker over attraktioner i Aalborg, hvorefter brugeren vælger de attraktioner vedkommende ønsker at besøge. Programmet vil ud fra disse punkter, beregne en kort rute, og undersøge om der er andre attraktioner, som ligger tæt på ruten, og spørge brugeren, om det kunne være interessant at besøge disse steder. Hvis ja, vil disse punkter også blive inkluderet. På den måde får brugeren selv lov til at skabe sig den mest interessante rute.

Resultatet bliver en liste over destinationerne, der står i rækkefølge, så turisten ved hvilken rækkefølge de skal besøge dem i, for at få den mest optimale rute.

Det næste afsnit vil omhandle de teorier, som beskriver de formler, algoritmer og metoder, der er brugt i løsningen.

I dette afsnit vil teorien, der ligger til grund for programmet, blive beskrevet, hvorunder tre hovedafsnit findes. Grafteori vil være det første afsnit, efterfulgt af vektorteori og positionsbestemmelse. Under grafteori vil et par forskellige algoritmer tages i brug, hvoraf den bedste match til programmet findes. I grafteorien vil Nearest neighbor, Dijkstra's algoritme og Double Minimum Spanning Tree algoritmerne blive beskrevet. I vektorteorien vil teorien bag udregning af afstanden mellem to knuder, eller en knude og en rute gøres rede for. Denne teori vil blive brugt i programmet til dannelsen af rute, og udregning af nærliggende attraktioner.

5.1 Grafteori

Grafteori er et afsnit i denne rapport, som omhandler en generel forklaring på grafteori, hvorefter teorien bag Nearest neighbor algoritmen, Dijkstra's algoritmen, Double Minimum Spanning Tree og Traveling salesman problem vil blive beskrevet. Alt dette beskrives, for at give et udgangspunkt for implementering af en hensigtsmæssig algoritme i programmet for denne rapport.

5.1.1 Begrebsbeskrivelse

En knude er et punkt på grafen. Knuder vil i den rapport sammenhæng være attraktioner.

En kant forbinder to knuder. Kanter kan vægtes, i denne rapport vil vægtingen af kanter være afstanden mellem de to knuder som kanten forbinder.

En path er en rute gennem et antal knuder, hvor startknuden ikke er endeknuden.

Et circuit er en rute gennem et antal knuder, hvor startknuden også er endeknuden.

Et Hamiltonian circuit er et circuit, hvori alle knuder er besøgt én gang, med undtagelse af slutpunktet, som skal være det samme som startpunktet.

I denne rapport søges et circuit, som går igennem alle knuder én gang, hvor en samlet vægting af kanterne (længden af ruten) bliver så kort som mulig. Dette er i grafteori også kaldet Traveling Saleman Problem. En umiddelbar løsning til TSP ville være at undersøge alle mulige Hamiltonian circuits, og finde den korteste. Problemet er, at når en graf med n knuder skal undersøges, skal der bruges $(n-1)!/2$ udregninger. Dette skyldes, at efter en startknude er valgt, vil der være $(n-1)$ mulige knuder som næste punkt, og derefter $(n-2)$ osv. Da der ikke er en bestemt retning på Hamiltonian circuits, vil der altid være to veje, hvor den ene beskriver den anden bare omvendt. Derfor fås $(n-1)!/2$. Med 25 knuder vil der være $(24)!/2$, som svarer til

$$3.1 * 10^{23}$$

forskellige Hamiltonian circuits. Hvis det antages, at det tager et nanosekund at udregne ét Hamiltonian circuit, vil dette tage cirka 10 millioner år, at udregne alle circuits, og finde den optimale. Derfor bruges algoritmer, som udregner en løsning som er tæt på den optimale. Nedenfor vil tre sådanne algoritmer blive beskrevet. [Rosen, 2011]

5.1.2 Nearest neighbour algoritmen

Fremgangsmetode:

Vælg en startknode.

Følg kanten med den laveste vægtning.

Check om der er flere ubesøgte knuder tilbage, hvis ja, gå tilbage til trin 2.

Gå tilbage til startknuden.

Fra den knude der behandles, skal kanten med den laveste værdi følges. Hvis alle knuder er besøgt, skal kanten der går tilbage til startknuden vælges.

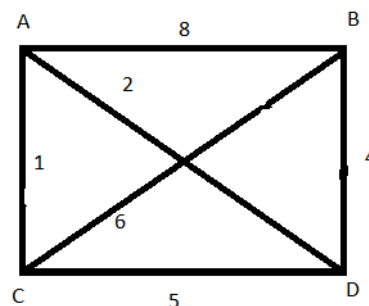
Eksempelvis i dette tilfælde (figur 5.1), er den endelige rutes længde, udregnet med NNA: $1+5+4+8+2 = 20$. NNA udregner ikke med sikkerhed en optimal rute, hvilket skyldes at den ikke tager højde for konsekvenserne af de skridt den tager. Selvom det første skridt er det korteste blandt de mulige, kan det i det lange løb godt ende med at blive en meget længere rute der bliver sammensat. En måde dette kan optimeres, er ved at køre NNA flere gange med forskellige startpunkter. Her kaldet en udvidet NNA. Nedenfor er en udvidet NNA beskrevet.

Fremgangsmetode:

Udfør NNA på en ny ikke testet startknode.

Er der knuder der ikke er testet som startknode, hvis ja, gå tilbage til trin 1.

Vælg det korteste circuit af de testede.



Figur 5.1: Hamiltonian Circuit.
Følger kanter med laveste værdier.

	A	B	C	D
A	-	8	1	2
B	8	-	6	3
C	1	6	-	5
D	2	4	5	-

De mulige ruter er:

$$ACDBA = 1 + 5 + 4 + 8 = 18$$

$$BDACB = 4 + 2 + 1 + 6 = 13$$

$$CADBC = 1 + 2 + 4 + 6 = 13$$

$$DACBD = 2 + 1 + 6 + 4 = 13$$

Den korteste rute er altså BDACB, hvilket er 5 kortere end den antagede rute. Den optimale circuit fundet med udvidet NNA er derfor denne rute. Dette tager standard NNA ikke højde for, da den starter i en valgt start-knode, og derefter følger kanten, med den derfra laveste værdi. NNA kræver ikke lige så mange beregninger som en brute force udregning kræver, og kan derfor bruges i praksis, dog er den ikke sikker på at finde den optimale rute.

5.1.3 Dijkstra's algoritme

Udover NNA, findes også Dijkstra's algoritme, hvor første trin er, at bestemme ende-knuden, og sætte dens distance til nul. Denne knude sættes til at være den første knude, som behandles. I det

en knude er checket færdig, vil denne knude markeres som "besøgt", og kanten med den mindste værdi følges, og næste knude markeres som "nuværende" knude. En kant bliver kun fulgt, hvis det er den korteste rute, tilregnet tidligere kanter. Problematikken med Dijkstra's algoritme i forhold til dette projekt er, at den checker den korteste rute fra start-knude til slut-knude, men den inkluderer ikke nødvendigvis alle knuder som oplyses. I det denne rapport er afgrænset til fugleflugtslinjer, vil Dijkstra's ikke være den optimale. Hvis en rute igennem en by, hvor der er tilregnet veje, stier og andre knuder, vil Dijkstra's være det bedste valg. Denne algoritme vil også være i brug ved den optimale løsning. Ved brug af Dijkstra's algoritme, vil den nuværende rute altid blive testet for, hvorvidt ruten der undersøges efter, er kortere eller længere end den hidtil korteste rute. Hvis den er kortere, vil denne rute blive sat som den hidtil korteste rute. [Mathispower4u, 2013]

/subsectionDouble Minimum Spanning Tree Double Minimum Spanning Tree er algoritmen der ligger til grund for Christofides algoritme, hvor DMST har tre trin:

Først oprettes et minimum spanning tree, som inkluderer alle knuder.

Alle kanter duplikeres.

Den korteste rute findes mellem disse kanter, hvor en knude kun besøges én gang.

Hvis der ikke er kanter til ubesøgte knuder, oprettes en "genvej" fra den nuværende knude til en ubesøgt knude [Nilsson, 2003].

Et minimum spanning tree findes ved at tage så få kanter som muligt, med den laveste vægtning, så alle knuder er besøgt blot én gang.

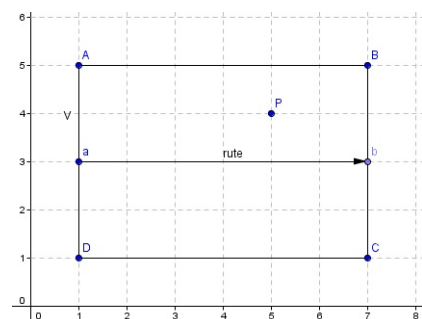
5.1.4 Opsummering

Da dette projekt forholder sig til en forholdsvis kort rute, men hovedsagligt tager udgangspunkt i en rute hvortil der kan tilføjes flere punkter, for at lave en interessant rute, er NNA valgt som den bedste kandidat. Dette skyldes, at Dijkstra's ikke nødvendigvis inkluderer alle knuder, da den finder en kort rute fra startpunkt til slutpunkt. DMST finder en mere optimal rute end NNA, men samtidig kræver den længere tid at udregne, og derfor blev NNA valgt som den optimale algoritme til dette projekt.

5.2 Vektorteori

Essensen i dette projekt er at finde en flerpunktsrute mellem nogle valgte attraktioner, hvor brugeren skal have mulighed for, at vælge nogle attraktioner til deres rute. Gruppen vil ikke diktere hvad en interessant rute er for brugeren, derfor skal de have muligheden for at vælge de foreslåede attraktioner til eller fra.

Der tages nu udgangspunkt i figur 5.2. En del af brugerens rute ligger fra attraktion a til attraktion b. Der skal nu tjekkes om der ligger andre attraktioner mellem afstanden fra a til b (eller AB), og med bredden AD hvor brugeren vil blive spurgt om denne attraktion skal tilføjes til ruten. AD er i projektets program sat til at være $V * 2$. Dette vil blive udregnet vha. vektorer Hvis der antages at punktet P er en attraktion som programmet skal tjekke, ligger



Figur 5.2: Eksempel på om en attraktion er indenfor punkt a og b.

denne inden for længden af ruten AB og bredden AD. Dette tjekkes med følgende formel:

$$0 < AP \cdot AB < AB \cdot AB \wedge 0 < AP \cdot AD < AD \cdot AD$$

Hvor prikproduktet af vektorerne AP og AB, skal være større end 0 og mindre end prikproduktet af vektorerne AB og AB. Det samme vil gælde for AD i stedet for AB.

Lad nu som om det de informationer der kendes er punkterne a og b, samt længden på vektor ab som vil være 6 og vektoren vil hedde:

$$\begin{bmatrix} 6 \\ 0 \end{bmatrix}$$

Først ønskes punktet A findes, som gøres ved først at finde tværvektoren. Tværvektoren findes ved at bytte 1. og 2. koordinat rund og ændre fortegn på første koordinaten:

$$\begin{matrix} a1 & -a2 \\ a2 & a1 \end{matrix}$$

Tværvektoren hedder:

$$\begin{bmatrix} 6 \\ 0 \end{bmatrix},$$

og har udgangs punkt fra punktet a.

I dette eksempel skal der søges efter ekstra attraktioner langs ruten, svarende til 1/3 af rutens længde. Så for at finde koordinaterne til punktet A, finder vi først en enhedsvektor for tværvektoren, dette gøres med formelen:

$$\vec{e} = \frac{1}{\|\vec{a}\|} * \vec{a}$$

Dette giver en vektor:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

som også har udgangspunkt fra punktet a. Som tidligere nævnt søges der efter ekstra attraktioner langs ruten, svarende til 1/3 af rutens længde, så enhedsvektoren multipliceres med to, hvilket giver en vektor:

$$\begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

Denne vektor lægges til koordinaterne til punktet a, hvilket vil give punktet:

$$A = \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}.$$

Punktet D vil så ledes findes ved at tage vektoren fra før og multiplicere med -2 og lægge punktet A til:

$$D = \begin{bmatrix} 0 \\ 2 \end{bmatrix} * -2 = \begin{bmatrix} 0 \\ -4 \end{bmatrix} + \begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Dog vil der først findes en vektor AP mellem punkterne A og P med formelen:

$$\overrightarrow{AP} = \begin{bmatrix} X2 - X1 \\ Y2 - Y1 \end{bmatrix}$$

Vektor AP: A(1,5) og P(5,4):

$$\overrightarrow{AP} = \begin{bmatrix} 5 - 1 \\ 4 - 5 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

Vektor AB er allerede kendt, da det er det samme som \overrightarrow{ab} .

For at projekte AP på AB skal følgende formel benyttes: [MatematikFessor, 2014]

$$b_a = \left(\frac{a \cdot b}{|a|^2} \right) \cdot a$$

Med denne formel vil vektoren b blive projekteret på vektoren a. I tælleren findes prikproduktet som kan findes ved at:

$$a \cdot b = \begin{bmatrix} X1 & X2 \\ Y1 & Y2 \end{bmatrix}$$

I nævneren findes længden på vektor a i anden, som kan regnes ved at sige:

$$\sqrt{ax^2 + ay^2}$$

Hvis der forsat tages udgangspunkt i eksemplet med figur 5.2, vil projektionen af AP på AB kunne ses på figur 5.3:

Prikproduktet af vektorerne:

$$\overrightarrow{AP} \cdot \overrightarrow{AB} = \begin{bmatrix} 4 & 6 \\ -1 & 0 \end{bmatrix} = 4 \cdot 6 + (-1) \cdot 0 = 24$$

Længden af AB opløftet i anden vil være:

$$\sqrt{6^2 + 0^2} = 6$$

Ud fra dette kan vektoren fra projektionen af AP på AB findes:

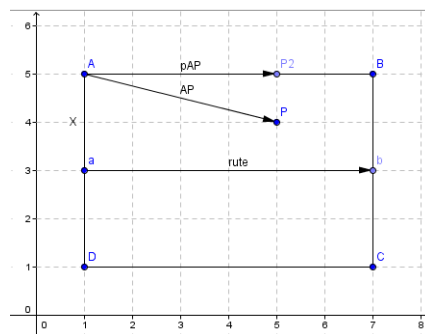
$$\frac{24}{36} \cdot \begin{bmatrix} 6 \\ 0 \end{bmatrix} \rightarrow \frac{24}{36} \cdot 6 \wedge \frac{24}{36} \cdot 0 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

Hvor resultatet vil give en ny vektor:

$$\begin{bmatrix} 4 \\ 0 \end{bmatrix},$$

som også vil have startpunkt i A. Hvis der igen tages udgangspunkt i formelen:

$$0 < AP \cdot AB < AB \cdot AB \wedge 0 < AP \cdot AD < AD \cdot AD$$



Figur 5.3: Fortsat eksempel på om en attraktion er indenfor punkt a og b.

overholder punktet P første del, og ovenstående metode skal derfor gentages med vektoren AD i stedet for AB, for matematisk at finde ud af om punktet ligger inden for den afsatte bredde og længden af ruten a til b. Ved udregning af projektionen af AP på AD vil den nye vektor hedde:

$$\begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$

Opsummering

I grafteorien blev der konkluderet, at Nearest Neighbor algoritmen er den mest egnede til programmet, da dette projekt forholder sig til udregning af en relativt kort rute, hvorved en interessant rute kan findes efterfølgende. Vektorteorien gjorde grund for udregning af, hvilke attraktioner der er nærliggende, så en interessant rute kan findes ud fra ruten der i første omgang bliver dannet af NNA.

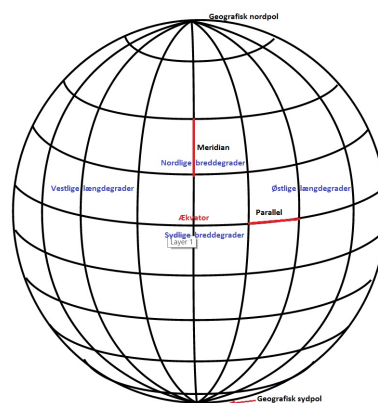
5.3 Positionsbestemmelse

I projektet bruges længde- og breddegrader til at udregne afstanden mellem attraktionerne. Dette sker kort sagt ved, at finde to koordinatsæt og bruge pythagoras' sætning til at udregne afstanden. I dette afsnit vil der være en kort forklaring om længde- og breddegrader, og hvordan gruppen har brugt dem i dette projekt.

Længdegrader er de lodrette streger der kan ses på globussen i figur X. De lodrette streger kaldes for en meridian, hvor disse vil have mindre afstand mellem hinanden i nærheden af de geografiske poler, i forhold til når de er tæt ved ækvator. De vandrette streger er breddegrader, hvor disse er kaldt parallel. Breddegradernes afstand ændre sig ikke på samme måde som længdegraderne. [Postamenter, 2014]

Ligesom på en cirkel er jorden delt op i 360 længdegrader. 0 grader vil ligge i observatoriet i Greenwich, som ligger i London området. De 360 grader er opdelt i 180 vestlige og 180 østlige længdegrader. Breddegrader begynder fra 0 grader ved ækvator og rammer 90 grader ved en af polerne, i henholdsvis nordlige- eller sydlige breddegrader. [Postamenter, 2014]

Der findes forskellige metoder til at notere koordinatsættet i form af længde- og breddegrader. I dette projekt bruges der et grade tal med decimaler, eksempelvis 57,12345. I dette system nævnes ikke om det er i sydlig, nordlig, vestlig eller østlig retning, men istedet er de sydlige breddegrader og vestlige længdegrader, negative tal. En anden måde at notere dette er at dele det op i grader, minutter og sekunder, hvor der går 60 sekunder på et minut og 60 minutter på en grad. Breddegraderne ligger fast på ca. 111 km pr. grad, mens længdegraderne variere da der bliver kortere afstand mellem meridianerne jo tættere de er på polerne. Med andre ord kan det siges, at hvis man er på 5. længdegrad, så er man 5 længdegrader fra Greenwich (0 grader), men afstanden i km for hver længdegrad ændre sig i forhold til hvilken breddegrad der undersøges. Denne afstand kan enten udregnes eller findes i en tabel. I dette projekt er det afstanden kigget op i en tabel [Humerfelt, 1999]. Som viser at der ved Aalborg, ved den 57. breddegrad, går 1012.87 meter per minut, altså $1012.87 \cdot 60 = 60772.2$ meter pr grad = 60.77 kilometer pr grad. [Postamenter, 2014]



Figur 5.4: Fortsat eksempel på om en attraktion er indenfor punkt a og b.

Implementering 6

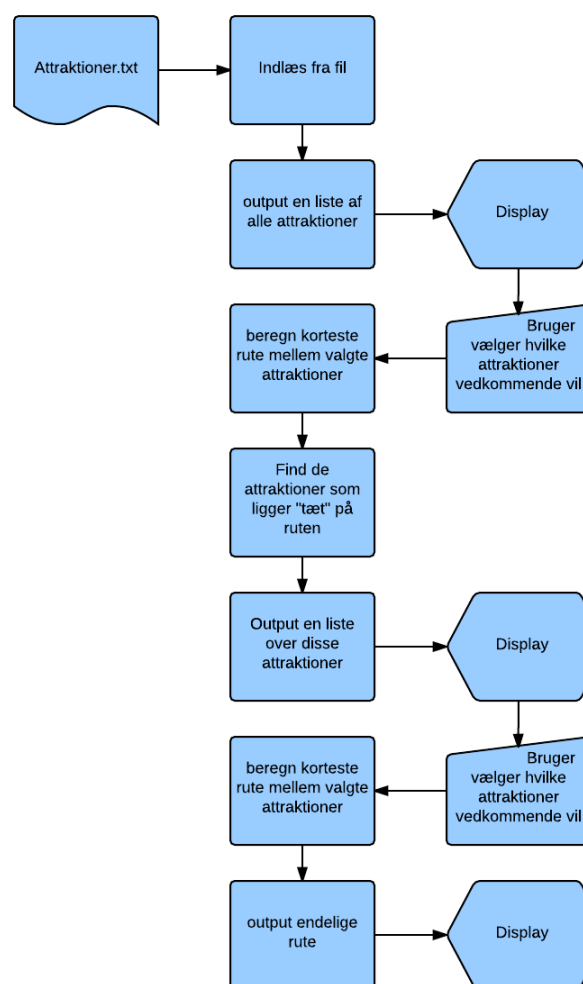
I dette afsnit vil programmet blive beskrevet, i form af en overordnet programbeskrivelse og en dybdegående forklaring af programmets funktioner. Derudover vil der blive beskrevet hvordan brugeren interagerer med programmet.

6.1 Programbeskrivelse

Programmet begynder med at indlæse alle de forudbestemte tilgængelige attraktioner fra en tekstfil. Alle disse attraktioner bliver vist på en liste for brugeren i kommandoprompten med numre ud for hver attraktion. Brugeren vælger, hvilke attraktioner vedkommende har lyst til at besøge, ved at indtaste attraktionens nummer, hvor attraktionen så vil blive tilføjet til en liste. Denne liste kører gennem programmet, og den korteste rute bestemmes. Programmet tjekker derefter, for nærliggende attraktioner der kan tilføjes til ruten, på samme måde som i begyndelsen da de valgte attraktioner til deres rute. Hvis brugeren vælger ekstra attraktioner til deres rute, kører programmet igen, og den nye rute beregnes. Når ruten er beregnet, vil ruten blive vist på skærmen, i den beregnede rækkefølge. Den samlede længde af ruten vil også blive vist.

6.2 Beskrivelse af structs

Til programmet bruges fire forskellige structs. Den første struct er "attraktion", som indeholder to strings, "navn" og "adresse", to doubles, "kmFraGreenwich" og "kmFraAekvator", og en int, "besoegt", der beskriver om attraktionen er besøgt eller ej. Denne struct bruges til at opdele og gemme de enkelte attraktioners data i programmet.



Figur 6.1: Flowchart over programmet

```
1 typedef struct {
    char navn[MAX_STRING];
3   char adresse[MAX_STRING];
    double kmFraGreenwich, kmFraAekvator;
5   int besøgt;
} attraktion;
```

Næste struct er "naboRute", som indeholder en double, "ruteLaengde", og et array af attraktioner, "rute". Denne struct bruges til at lave ruter, hvor "rutelaengde" er længden af ruten og arrayet indeholder rækkefølgen af attraktioner på ruten.

```
typedef struct {
2   double ruteLaengde;
    attraktion rute[ANTAL_ATTRAKTIONER];
4 } naboRute;
```

Næste struct er "kant", som indeholder to attraktioner, "startAttraktion" og "slutAttraktion", og en double, "laengde". Denne struct bliver brugt til at gemme data omkring afstanden mellem to punkter.

```
typedef struct {
2   attraktion start;
    attraktion slut;
4   double laengde;
} kant;
```

Sidste struct er "vektor", som indeholder to doubles, "x" og "y". Denne struct bruges til at gemme x og y koordinater for vektorer og i nogle tilfælde punkter.

```
1 typedef struct {
    double x, y;
3 } vektor;
```

6.3 Beskrivelse af funktioner

Attraktionerne til programmet bliver indlæst fra en .txt fil når programmet køres. I programmet bliver filen indlæst i en funktion som hedder initialiserAttraktioner. Hvis filen ikke er tom vil elementerne, vha. funktionen fscanf, blive indlæst i grupper af 4, hvor de bliver indlæst som "attraktion" hvilket er defineret som et struct. Hvis filen er tom, vil en advarsel blive vist i prompten og lukke programmet ned. Når alt information er indlæst fra filen, og den ikke længere er nødvendig, lukkes filen.

```
1 void initialiserAttraktioner(attraktion *attraktioner){
    FILE *input_file_pointer;
3   int i = 0;
    double lndgrad;
5   double brdgrad;

7   input_file_pointer = fopen("attraktioner.txt", "r");
```

```

9  if(input_file_pointer != NULL){
    while(fscanf(input_file_pointer, " %s %s %lf %lf", attraktioner[i].navn, attraktioner[i].adresse,
        &brdgrad, &lndgrad) == 4){
11     attraktioner[i].kmFraGreenwich = lndgrad * KM_PR_LNDGRAD;
        attraktioner[i].kmFraAekvator = brdgrad * KM_PR_BRDGRAD;
13     attraktioner[i].besoegt = 0;
        i++;
15     }
    }else{
17     printf("kunne ikke aabne fil\n"); exit(1);
    }
19     fclose(input_file_pointer);
}

```

Funktionen "udregn_kanter" bruges til at udregne distancen mellem punkterne. Beregningen af distancerne sker vha. "bereg_n_dist", ved at sende en start og slut attraktion. Derefter bliver start- og slut attraktionen plus længden imellem dem lagt ind som en kant i kanter arrayet. Dette gøres i en for-løkke i en anden for-løkke, hvor der for hvert punkt bliver der udregnet distancen til de punkter der ikke allerede er blevet oprettet en kant for, i en tidligere iteration af løkkerne. For hver kant der bliver oprettet, oprettes en ekstra kant som har samme længde, men har modsat start og slut attraktion. Så hvis der oprettes en kant fra x til y, vil der også blive oprettet en kant fra y til x, med samme længde. Kanterne bliver tilføjet til kantarrayet, gennem den pointer der er inputparameter til funktionen, så kanterne kan blive tilgået fra resten af programmet.

```

void udregn_kanter(attraktion *attraktioner, kant *kanter)
2 {
    int i;
    int y;
    int indexTilKanter = 0;
    for (i = 0; i < ANTAL_ATTRAKTIONER; i++)
    {
        for (y = i; y < ANTAL_ATTRAKTIONER; y++)
        {
10         if (strcmp(attraktioner[i].navn, attraktioner[y].navn) != 0)
            {
12             kant k, j;
                k.start = attraktioner[i];
                j.slut = k.start;
14             k.slut = attraktioner[y];
                j.start = k.slut;
                k.laengde = beregn_dist(k.start, k.slut);
18             j.laengde = k.laengde;
                kanter[indexTilKanter] = k;
                kanter[indexTilKanter+1] = j;
20             indexTilKanter += 2;
            }
22         }
    }
24 }
}

```

Funktionen "bereg_n_dist" er en implementering af pythagoras sætning til at finde længden mellem 2 punkter. Funktionen returnere den beregnede distance mellem de to attraktioner.

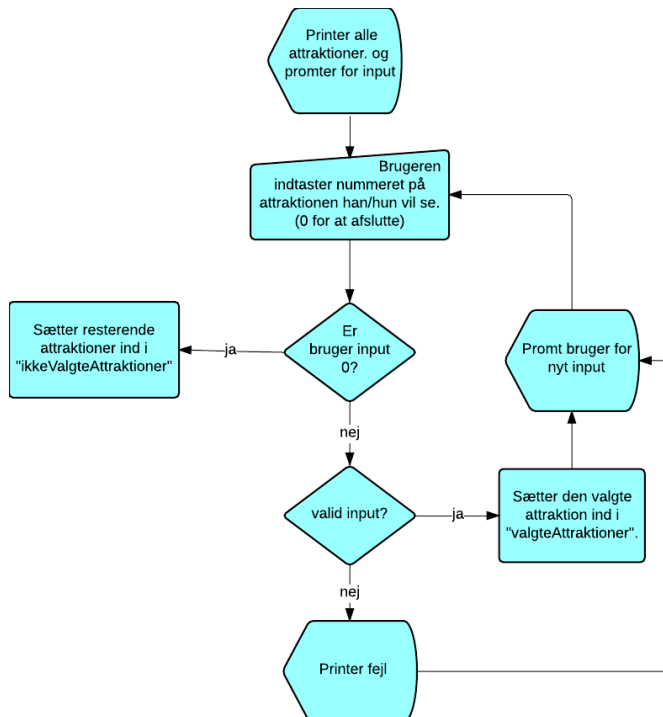
```

1 double beregn_dist(attraktion startAttraktion, attraktion slutAttraktion)
{
3     return sqrt(pow(startAttraktion.kmFraGreenwich - slutAttraktion.kmFraGreenwich, 2) +
        pow(startAttraktion.kmFraAekvator - slutAttraktion.kmFraAekvator, 2));
}

```

5 }

Brugeren bliver præsenteret for en liste over alle tilgængelige attraktioner fra inputparameteren, og attraktionernes tilhørende nummer i funktionen "valgafAttraktioner. Brugeren bliver bedt om at indtaste attraktionernes matchende numre, hvilket vil blive tilføjet til en list. Taster brugeren 0, bliver brugeren præsenteret for sine valg, og beregninger af den korteste rute igangsættes. Programmet returnerer listen af valgte attraktioner.



Figur 6.2: Flowchart over "valgafAttraktioner" funktionen

```

1 void valgafAttraktioner(attraktion *attraktioner, attraktion *valgteAttraktioner, int
  *antalValgteAttraktioner, attraktion *ikkeValgteAttraktioner){
  int i = 0, j = 0, k = 0, l = 0, m = 0, n = 0, valgt = 0, y = 0;
3  int opretNy = 1;

5  printf("Rutevejledning - A401 - P1\n");
  printf("Dette program vil, ud fra dine onskede attraktioner, bestemme den korteste rute.\n");
7  printf("Du kan se de attraktioner du kan vælge her, den foerske du vælger, vil være der hvor du
    starter:\n");

9  int halvdelen = ANTAL_ATTRAKTIONER/2;

11 for(i = 0; i < halvdelen; i++){
    printf("%d: %-40s \t%d: %s\n", i+1, attraktioner[i].navn, i+halvdelen+1,
13     attraktioner[i+halvdelen].navn);
  }

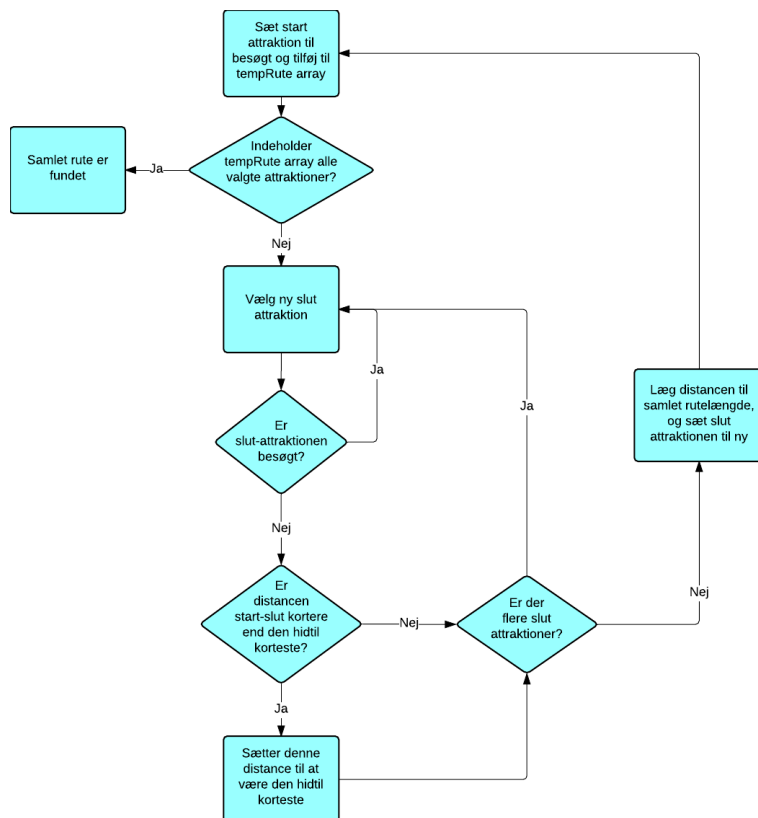
15 printf("Vaelg de attraktioner du ønsker at se ved at skrive det tilhoerende tal.\n");
  printf("Vaelg et tal (svarende til en attraktion) af gangen og tryk enter efter hver indtastet tal.\n");
17 printf("Tast 0, efterfulgt af enter, for at gaa videre. Indtast ikke samme tal 2 gange.\n");
  do{
19     opretNy = 1;
    if (scanf("%d", &k) == 1){
21     for (y = 0; y < j; y++)
        {

```



```
23     if (strcmp(valgteAttraktioner[y].navn, attraktioner[k-1].navn) == 0)
24     {
25         printf("Du har allerede indtastet denne attraktion. Proev igen.\n");
26         opretNy = 0;
27     }
28 }
29 if (k == 0)
30     break;
31 else if (k > ANTAL_ATTRAKTIONER || k < 0)
32     printf("Tallet svarer ikke til en attraktion\n");
33 else if (opretNy && (k <= ANTAL_ATTRAKTIONER))
34 {
35     valgteAttraktioner[j] = attraktioner[k-1];
36     printf("Tilfoejet attraktion: %s\n", attraktioner[k-1].navn);
37     j++;
38 }
39 }
40 else
41 {
42     printf("Fejlindtastning - proev igen\n");
43     char e[MAX_STRING];
44     scanf("%s", e);
45     k = 1;
46 }
47 } while(j < ANTAL_ATTRAKTIONER && k != 0);
48
49
50
51 for (l = 0; l < ANTAL_ATTRAKTIONER; ++l)
52 {
53     valgt = 0;
54     for (m = 0; m < j; ++m)
55     {
56         if(strcmp(attraktioner[l].navn, valgteAttraktioner[m].navn) == 0){
57             m = j;
58             valgt = 1;
59         }
60     }
61     if(valgt != 1){
62         ikkeValgteAttraktioner[n] = attraktioner[l];
63         n++;
64     }
65 }
66 *antalValgteAttraktioner = j;
67 }
```

Funktionen "findNaboRute" benytter NNA (Nearest Neighbour Algorithm, eller Nærmeste Nabo Algoritme), til at finde den korteste rute mellem en række attraktioner for et bestemt startsted, der er angivet i inputparametrene. Efter at have fundet den korteste rute ud fra NNA som beskrevet i teoriafsnittet 5.1.1 om grafteori, returnere den et array med ruten og distancen for denne rute.



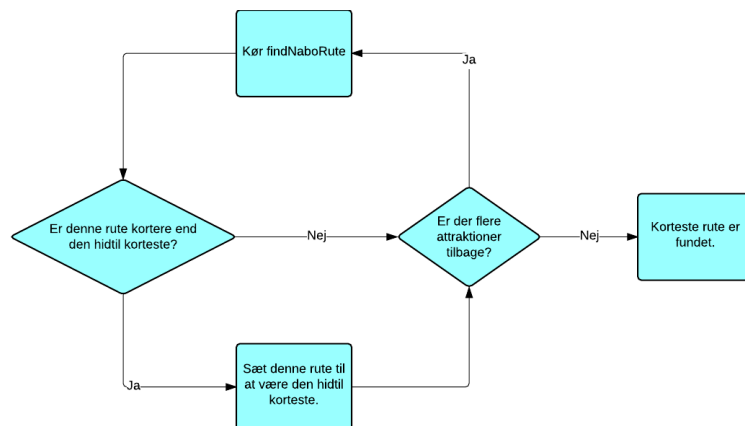
Figur 6.3: Flowchart over "findNaboRute" funktionen

```

1 void findNaboRute(attraktion *valgteAttraktioner, int antalValgteAttraktioner, attraktion
  *startAttraktion, kant *kanter, attraktion **tempRute, double *ruteLaengde){
2   int i = 0;
3   double lavesteLaengde = 10000;
4   *ruteLaengde = 0;
5
6   tempRute[i] = startAttraktion;
7
8   for(i = 0; i < antalValgteAttraktioner-1; ++i){ /*da der f.eks. kun er 4 kanter imellem 5 punkter*/
9     tempRute[i]->besoegt = 1;
10    int j = 0;
11    for (j = 0; j < antalValgteAttraktioner; ++j)
12    {
13      if(valgteAttraktioner[j].besoegt != 1 && findDist(*tempRute[i], valgteAttraktioner[j], kanter) <
        lavesteLaengde){
14        lavesteLaengde = findDist(*tempRute[i], valgteAttraktioner[j], kanter);
15        tempRute[i+1] = &valgteAttraktioner[j];
16      }
17    }
18    *ruteLaengde += lavesteLaengde;
19    lavesteLaengde = 10000;
20  }
21  tempRute[antalValgteAttraktioner] = startAttraktion;
22  *ruteLaengde += findDist(*tempRute[antalValgteAttraktioner-1], *tempRute[antalValgteAttraktioner],
    kanter);
23 }
  
```

"findKortesteNaboRute" benytter funktionen "findNaboRute", til at finde ud af hvilket startsted der giver den korteste rute. Dette gøres ved at sætte en variable til en stor værdi, som rutelængden ikke vil gå over, og opdatere den hvis "findNaboRute" returnerer en distance for ruten gennem de givne attraktioner med en given start attraktion, der er lavere end de foregående ruter. "findKortesteNaboRute" returnerer et array med den korteste rute, og den samlede længde

af denne rute.



Figur 6.4: Flowchart over “findKortesteNaboRute” funktionen”

```

1 void findKortesteNaboRute(attraktion *valgteAttraktioner, int antalValgteAttraktioner, attraktion
   *ruteAttraktioner, kant *kanter, double *samletLaengde){
   /* input er valgteAttraktioner arrayet, og kanter arrayet*/
3  /*output er ruteAttraktioner og samletLaengde*/
   double ruteLaengde;
5  attraktion *tempRute[antalValgteAttraktioner+1];
   *samletLaengde = 100000;

7
   int i;
9   int h;
   int j;
11  for (i = 0; i < antalValgteAttraktioner; ++i)
   {
13     for (h = 0; h < antalValgteAttraktioner; ++h)
       {
15         valgteAttraktioner[h].besoegt = 0;
       }
17     findNaboRute(valgteAttraktioner, antalValgteAttraktioner, &valgteAttraktioner[i], kanter, tempRute,
       &ruteLaengde);
       if(ruteLaengde < *samletLaengde){
19         *samletLaengde = ruteLaengde;
         for (j = 0; j < antalValgteAttraktioner+1; ++j)
21         {
           ruteAttraktioner[j] = *tempRute[j];
23         }
       }
25     }
   }
}

```

Funktionen "findDist" gennemgår alle kanter der allerede er oprettet, og returnere distancen mellem to attraktioner, som er sendt med kaldet af funktionen, uden at skulle beregne den igen.

```

double findDist(attraktion start, attraktion slut, kant *kanter){
2   int i;
   for (i = 0; i < ANTAL_KANTER; ++i)
4   {
       if(strcmp(kanter[i].start.navn, start.navn) == 0 && strcmp(kanter[i].slut.navn, slut.navn) == 0){
6         return kanter[i].laengde;
       }
8   }
   printf("Kunne ikke finde passende kant\n"); exit(0);
10 }

```

Funktionen "attraktionErTilfoejt" bruges til at finde ud af, om en attraktion allerede er tilføjet til ens liste over ekstra attraktioner, og returnere enten true eller false.

```
int attraktionErTilfoejt(attraktion *ekstraAttraktioner, int antalEkstraAttraktioner, attraktion
    attraktionAtTilfoeje){
2   int i;
   for (i = 0; i < antalEkstraAttraktioner; ++i)
4   {
       if(strcmp(ekstraAttraktioner[i].navn, attraktionAtTilfoeje.navn) == 0){
6           return 1;
       }
8   }
   return 0;
10 }
```

Funktionen "prikProdukt" bruges til at finde prikproduktet mellem to vektorer, hvilket også er det funktionen returnere.

```
double prikProdukt(vektor vektor1, vektor vektor2){
2   return (vektor1.x * vektor2.x) + (vektor1.y * vektor2.y);
}
```

Funktionen "findEkstraAttraktionerFirkant" benytter beregningerne fra teori afsnittet "Vektorteori", til at finde ud af om der findes en evt. interessant attraktion på brugerens rute. Der tjekkes om attraktionAtTilfoeje ligger inden for en bestemt distance til linjen mellem start og slut punkter. Hvis attraktionAtTilfoeje ligger indenfor, bliver den lagt i ekstraAttraktioner arrayet.

```
1 void findEkstraAttraktionerFirkant(attraktion startAttraktion, attraktion slutAttraktion, kant *kanter,
    attraktion attraktionAtTilfoeje, double maxDist,
    attraktion *ekstraAttraktioner, int *antalEkstraAttraktioner){
3   double vektorLaengde;
   vektor ruteVektor, ruteVinkelretVektor, ruteEnhedsVinkelretVektor, mainHjoerne, side1Vektor,
       side2Vektor, punktVektor;
5
   vektorLaengde = findDist(startAttraktion, slutAttraktion, kanter);
7   ruteVektor.x = slutAttraktion.kmFraGreenwich - startAttraktion.kmFraGreenwich;
   ruteVektor.y = slutAttraktion.kmFraAekvator - startAttraktion.kmFraAekvator;
9   ruteVinkelretVektor.x = -ruteVektor.y;
   ruteVinkelretVektor.y = ruteVektor.x;
11  ruteEnhedsVinkelretVektor.x = ruteVinkelretVektor.x / vektorLaengde;
   ruteEnhedsVinkelretVektor.y = ruteVinkelretVektor.y / vektorLaengde;
13  mainHjoerne.x = startAttraktion.kmFraGreenwich + ruteEnhedsVinkelretVektor.x * maxDist;
   mainHjoerne.y = startAttraktion.kmFraAekvator + ruteEnhedsVinkelretVektor.y * maxDist;
15  side1Vektor.x = ruteVektor.x;
   side1Vektor.y = ruteVektor.y;
17  side2Vektor.x = -2 * ruteEnhedsVinkelretVektor.x * maxDist;
   side2Vektor.y = -2 * ruteEnhedsVinkelretVektor.y * maxDist;
19
   punktVektor.x = attraktionAtTilfoeje.kmFraGreenwich - mainHjoerne.x;
21  punktVektor.y = attraktionAtTilfoeje.kmFraAekvator - mainHjoerne.y;
   if(0 < prikProdukt(punktVektor, side1Vektor) && prikProdukt(punktVektor, side1Vektor) <
       prikProdukt(side1Vektor, side1Vektor) &&
23  0 < prikProdukt(punktVektor, side2Vektor) && prikProdukt(punktVektor, side2Vektor) <
       prikProdukt(side2Vektor, side2Vektor)){
       ekstraAttraktioner[*antalEkstraAttraktioner] = attraktionAtTilfoeje;
```

```

25     *antalEkstraAttraktioner += 1;
    }
27 }

```

For at foreslå ekstra attraktioner til den valgte rute, bruges funktionen "findEkstraAttraktioner", hvor dette vil udgøre den interessante rute. Dette gøres den ved først at bruge funktionen "findDist", til at finde ikke valgte attraktioner inden for en bestemt distance af de valgte attraktioner. Er en attraktion ikke inden for denne radius, bruges funktionen "findEkstraAttraktionerFirkant" for at finde ud af, om attraktionen ligger tæt på ruten mellem to attraktioner. De attraktioner der enten er inden for den bestemte distance af enten attraktionerne eller ruterne derimellem, tilføjes til et array der returneres fra funktionen.

```

1 void findEkstraAttraktioner(attraktion *ruteAttraktioner, int *antalValgteAttraktioner,
    kant *kanter, attraktion *ikkeValgteAttraktioner, double maxDist, attraktion
    *ekstraAttraktioner, int *antalEkstraAttraktioner){
3
4     int i, j, antalIkkeValgteAttraktioner = ANTAL_ATTRAKTIONER - *antalValgteAttraktioner;
5
6     for (i = 0; i < *antalValgteAttraktioner; ++i)
7     {
8         for (j = 0; j < antalIkkeValgteAttraktioner; ++j)
9         {
10
11             if(attraktionErTilfoejet(ekstraAttraktioner, *antalEkstraAttraktioner, ikkeValgteAttraktioner[j])){
12                 }else if(findDist(ruteAttraktioner[i], ikkeValgteAttraktioner[j], kanter) < maxDist ||
13                     findDist(ruteAttraktioner[i+1], ikkeValgteAttraktioner[j], kanter) < maxDist){
14                         ekstraAttraktioner[*antalEkstraAttraktioner] = ikkeValgteAttraktioner[j];
15                         *antalEkstraAttraktioner += 1;
16                     }else{
17                         findEkstraAttraktionerFirkant(ruteAttraktioner[i], ruteAttraktioner[i+1],
18                             kanter, ikkeValgteAttraktioner[j], maxDist, ekstraAttraktioner,
19                             antalEkstraAttraktioner);
20                     }
21             }
22         }
23     }
24 }

```

Funktionen "aendre_startsted", tager en rute som inputparameter, sammen med det startsted der ønskes. Funktionen laver derefter et nyt array, der har det nye startsted som første og sidste element, så ruten starter i startstedet, og vender tilbage dertil. Dette array er hvad funktionen returnere.

```

void aendre_startsted(attraktion *ruten, attraktion nytStartSted, int antalAttraktioner, attraktion
    *outputRute)
2 {
3     int i = 0, startStedIndex = 0;
4
5     for (i = 0; i < antalAttraktioner; i++)
6     {
7         if (strcmp(ruten[i].navn, nytStartSted.navn) == 0)
8             startStedIndex = i;
9     }
10
11     for (i = 0; i < antalAttraktioner; i++)
12     {
13
14         if (startStedIndex == antalAttraktioner-1)
15         {

```

```
16     startStedIndex = 1;
    outputRute[i] = ruten[0];
18 }
    else
20 {
    outputRute[i] = ruten[startStedIndex];
22     startStedIndex++;
    }
24 }
}
```

For at sikre at programmet kører stabilt, og laver de korrekte udregninger, testes programmet for forskellige testcases. Der er forskellige metoder at teste dette på, hvoraf testen i denne rapport er blackbox testing. Denne testtype behandler en mængde testcases, som beskrevet i tabellen, hvor et bestemt input har et forventet output, og programmet testes derefter. Hvis det forventede output stemmer overens med det reelle output fra programmet, kan testen konkluderes succesfuld. En anden testtype kan være CU-test, som er et indlejret test-system, hvor testcases bliver oprettet som kode i programmet. Derefter vil testen blive kørt ved compile-time og resultater fra testcases bliver printet ud. Begrundelsen for brug af blackbox testing er, at der i dette program bliver printet gennem processen, og alle valg i programmet bliver foretaget af brugeren. Blackbox testing vil derfor give et godt billede, af om testen og input fra brugeren stemmer overens.

7.1 Testcases

Casene er opbygget således:

Case 1	Korrekt valgt af attraktioner	Korrekt tilføjelse af interessante punkter
Case 2	Korrekt valgt af attraktioner	For mange tilføjelser af interessante punkter
Case 3	Korrekt valgt af attraktioner	Forkert input-type ved tilføjelser af interessante punkter
Case 4	Korrekt valgt af attraktioner	Ingen tilføjelse af interessante punkter (input "0")
Case 5	Valg af flere attraktioner end muligt	Forventer ikke prompt for andet input
Case 6	Valg af samme attraktion flere gange	Forventer ikke prompt for andet input
Case 7	Forkert input-type til valg af attraktioner	Forventer ikke prompt for andet input
Case 8	Intet valg af attraktion (første input "0")	Forventer ikke prompt for andet input

Case 1:

I denne første testcase, vil inputtet til valg af attraktioner være 1, 5 og 9. Ved brug af disse tal, vil et forventet output være Aalborgslot for 1, Springerens - Maritimt Oplevelsescenter for 5 og Nordkraft for 9. Herefter ville valget af attraktioner afsluttes, ved input 0. Output ved første del af testcasen blev følgende: "Tilføjet attraktion: Aalborgslot", "Tilføjet attraktion: Springerens - Maritimt Oplevelsescenter" og "Tilføjet attraktion: Nordkraft". Herved gav den første del et korrekt output. Ved afsluttelse af valg af attraktion, tilføjes disse attraktioner til

ruten, og næste trin er tilføjelse af interessante, nærliggende attraktioner. Eftersom de valgte attraktioner alle ligger tæt på havnen i Aalborg, vil andre interessante attraktioner være Utzon Centeret, Havnefronten og Friis, da disse alle ligger tæt på en tiltænkt rute fra Nordkraft til Springereren, hvor Aalborg Slot også besøges.

Her foreslår programmet følgende: 1: Utzon_Centeret, 2: Friis_Aalborg_Citycenter og 3: Havnefronten. Dette er et korrekt output efter de attraktioner der blev valgt. Disse er alle tre inden for en afstand af 100 meter fra enten den nuværende rute eller de valgte attraktioner. Efterfølgende skal brugeren selv vælge, om han vil tilføje disse attraktioner til ruten. I dette tilfælde bliver inputtet 1 og 2, for tilføjelse af Utzon_Centeret og Friis_Aalborg_Citycenter. Outputttet blev "Tilføjet attraktion: Utzon_Centeret" og "Tilføjet attraktion: Friis_Aalborg_Citycenter". Dette stemmer overens med det forventede output, og tilføjelsen afsluttes med input 0. Herefter vil ruten blive dannet, og alle attraktioner valgt vil blive printet ud som "Din rute". Heraf vil der vises Aalborg Slot, Springereren, Nordkraft, Utzon Centeret og Friis. Disse vil sorteres efter hvornår på ruten de besøges, hvor startpunktet vil blive printet dobbelt, som både start-attraktion og slut-attraktion. Siden startattraktionen er Aalborg Slot, skal ruten blive Aalborg Slot, Utzon_Centeret, Friis_Aalborg_Citycenter, Nordkraft, Springereren - _Maritimt_Oplevelsescenter og Aalborg Slot. Dette er også tilfældet, da outputttet er mægen til det forventede:

Din rute:

Aalborg Slot

Utzon_Centeret

Friis_Aalborg_Citycenter

Nordkraft

Springeren - _Maritimt_Oplevelsescenter

Aalborg Slot

Herefter er der også et output der beskriver rutens længde, hvor et forventet resultat er udregnet gennem movable-type.co.uk, hvilket afrundet er 5.59km. Ifølge outputtet er længden 5.61km, hvilket er omkring 20 meter fra det forventede resultat. Dette er et fint resultat, som viser, at programmet i dette tilfælde har en fejlberegning på 0.31%. Denne fejlmargen er fin, da tallene i dette eksempel er afrundet.

Case 2:

Der bruges samme input i denne case, derfor vil testen være den samme, indtil tilføjelsen af interessante attraktioner skal have input. Dette input testes med et input der er højere end antallet af forslag, hvilket i dette tilfælde vil være 4. Her kommer en fejlmelding fra programmet: "Tallet svarer ikke til en attraktion", og der promptes efter nyt validt input. Ved indtastning af samme tal flere gange, kommer den forventede fejlmelding "Du har allerede indtastet denne attraktion. Proev igen".

Case 3:

Ligesom i case 2, er inputtet det samme indtil tilføjelsen af interessante attraktioner, hvor der i denne testcase testes for input af bogstaver, tegn og ord. I dette tilfælde vil "a", "!" og "test" alle tre printe "Fejlindtastning – prøv igen", og derefter promte efter nyt validt input. I en tidligere version printede programmet denne sætning for hvert tegn og bogstav inputtet bestod af. Så i "test" blev der printet fire "Fejlindstastning – prøv igen".

Case 4:

Igen her blev testen udført med det samme input som i case 2 bortset fra, at inputtet til tilføjelsen

af interessante attraktioner vil være "0", for afsluttelse af ruten uden tilføjelse af attraktioner. Her kører programmet videre, og giver den endelige rute:

Din rute:

Aalborghus_Slot

Nordkraft

Springeren_-_Maritimt_Oplevelsescenter

Aalborghus_Slot

Rutens længde ville forventet afrundet være 5.54 km, og outputtet fra programmet siger 5.56 km, som igen er omkring 20 meter længere.

Case 5:

Ved første prompt viser den antallet af attraktioner, og hvis alle attraktioner vælges, bliver der ikke promptet for en attraktion ud over det maksimale antal af attraktioner. Dette vil sørge for, at en bruger ikke kan vælge flere attraktioner end databasen er tilskrevet. Programmet vil derfor fortsætte videre til valg af alle attraktioner.

Case 6:

I denne testcase vil det første prompt testes for, hvorvidt det er muligt at indtaste den samme attraktion flere gange. Forventningen er, at en fejlmelding forekommer ved mere end én indtastning for samme attraktion. Ved indtastning af 1 to gange i træk, kom fejlmeldingen "Du har allerede indtastet denne attraktion. Proev igen". Derefter testes for, hvorvidt der kan skrives to forskellige tal, hvoraf det første bliver skrevet to gange, med det andet tal i mellem, dvs. 1, 2 og 1. Her forventes samme fejlmelding ved anden indtastning af 1 to gange i træk. Programmet printede den samme fejlmelding.

Case 7:

I testcase 7 vil det første prompt igen blive testet, denne gang for input af tegn, bogstaver og ord. Igen testes med "a", "!" og "test". Her forventes samme resultat som i testcase 3, hvor programmet i dette tilfælde vil printe fejlmeldingen "Fejlindtastning – prøv igen". Her var resultatet som forventet, i alle tre tilfælde blev der printet "Fejlindtastning – prøv igen". Ved indtastning af et tal højere end antallet af attraktioner, forventes fejlmeldingen "Tallet svarer ikke til en attraktion". Dette er også tilfældet, da programmet giver den rigtige fejlmelding.

Case 8:

I denne case testes programmet for intet valg af attraktioner, ved at første input er 0. Forventningen i denne test er, at programmet blot afsluttes. Dette er også tilfældet, dog havde en rettelse været nødvendig, da det originale program blot udskrev en rute uden attraktioner, med en afstand på 100.000 km.

Opsamling

Efter disse testcases kan det påvises, at programmet kører som planlagt, efter et par rettelser. Der er testet for, hvorvidt programmet giver en rute ved korrekt input, muligheder for forkert input, hvorvidt programmet giver en korrekt rutelængde, samt om attraktionerne tilføjet som interessante attraktioner er korrekte. Brugerens skulle ikke have mulighed for at give forkert input, rutelængden er med minimal fejlagtighed, korrekte interessante attraktioner bliver tilføjet, og ved et korrekt input vil en rute altid blive oprettet.

Diskussion 8

I dette kapitel vil gruppen opsamle på de beslutninger, som er blevet truffet gennem projektet. Samtidig vil gruppen se på fejlkilder der er opstået og hvilken indflydelse disse fejlkilder har haft på projektet.

8.1 Spørgeskema

Tidligt i projektet sendte gruppen et spørgeskema ud på deres facebookprofiler, som i alt gav 60 besvarelser. Da gruppen delte det på deres facebookprofiler, ville det betyde at gennemsnitsalderen på respondenterne, ville være relativ lav. Gruppen havde en række spørgsmål som vi gerne ville have svar på, og derfor blev spørgeskemaet opbygget af mange underspørgsmål, som var blevet lavet i forhold til den initierende problemstilling. Spørgeskemaet konstaterede nogle problemer, som fx at turisterne generelt havde svært ved at finde rundt, og foretrak en interessant rute, frem for en hurtig rute.

Turister var sat som ressourceperson i interessentanalysen, da det var dem som programmet var rettet imod, dog mener gruppen at spørgeskemaet blev sendt for hurtigt ud, og uden eftertanke. Med det menes der, at mange af spørgsmålene ikke var ordentlig formuleret. Herunder kan der ses et eksempel på et dårligt formuleret spørgsmål, som gruppen sendte ud:

- “Har du nogensinde haft problemer med at finde vej på din storbyferie?”

Dette eksempel, mener vi, er dårligt formuleret, da det er alt for åbenlyst hvad brugeren vil svare, for hvem har ikke haft problemer på en ferie? Her burde vi have været mere specifikke i spørgsmålet, som fx “I hvilke situationer har du haft problemer med at finde vej i en storby?”, med mulighed for at skrive eget svar i stedet for valgmuligheder.

Selve projektet er rettet mod turister i Aalborg, hvor 41% af turisterne er udenlandske, så da gruppen havde valgt at skrive spørgeskemaet på dansk, er det ikke hele målgruppen der rammes. Samtidig med dette, var spørgeskemaet som tidligere nævnt, delt på gruppemedlemmernes deres facebookprofiler, hvilket også begrænser respondentgruppen. De fleste respondenter vil derfor være dansktalende, hvilket i denne situation ikke var et problem, idet spørgeskemaet var skrevet på dansk. For at få flere respondenter, og specielt udenlandske respondenter, burde gruppen have delt spørgeskemaet flere steder end bare på Facebook, samt lavet spørgeskemaet på både engelsk og dansk.

Gruppen skulle have brugt mere energi/tid på spørgeskemaet, og med eftertanke burde vi nok have udsendt et nyt spørgeskema. Selve spørgsmålene skulle være bedre gennemtænkte og ikke så ledende som de var blevet skrevet. Ud fra spørgeskemaet blev der dog konstateret et problem, som projektet kunne tage udgangspunkt i. Respondentgruppen er dog relativ lille, og det ville

have været bedre at få en både større, men også bredere respondentgruppe. Med ordet bredere mener gruppen at der ønskes både respondenter fra ind- og udland i forskellige aldre.

8.2 Interview

I dette projekts interessentanalyse blev turistbureauer, i dette tilfælde VisitAalborg, sat som ressourceperson, da de kunne bidrage med informationer om turisme i Aalborg. Gruppen besluttede derfor at skaffe et interview med VisitAalborg, hvor gruppen fik fat i Lars Bech og Kim Mikael Jensen, som gerne ville stille op til et interview.

Gruppen havde udarbejdet en interviewguide, som var lavet i punktform, der beskrev hvilke tanker vi havde om emnet, og hvilke spørgsmål vi gerne ville stille Lars og Kim. Interviewet var planlagt som et ustruktureret interview, så hvis Lars/Kim ikke var meget for at snakke, eller ikke kom frem med det, vi efterspurgte, ville vi kunne spørge mere ind til emnet. Dette var dog ikke tilfældet med Lars, da han snakkede rigtig meget. Han førte ofte interviewet i en anden retning, end hvad vi havde planlagt. Eftersom vi i gruppen ikke rigtig havde lavet et interview før, var interviewerne ikke så gode til at stoppe ham, når han tog spørgsmålet i en anden retning. Dette gjorde at interviewet udviklede sig til, at Lars nærmest tog styringen af interviewet, men det gav mulighed for at få information, vi ikke selv havde overvejet, forud for interviewet.

Ud fra interviewet med Lars og Kim, blev gruppen klogere på hvilke turister der oftest besøger Aalborg, og hvilke attraktioner der er populære i Aalborg. Lars og Kim virkede interesserede i projektet, de havde dog tidligere arbejdet med en elektronisk løsning, men de var blevet nødsaget til at gå tilbage til kort og brochure.

Interviewet, var ligesom spørgeskemaet, udført uden den store eftertanke. Dette gjorde, at interviewguiden ikke blev så god, som gruppen havde håbet, og vi skulle i stedet have ventet til lidt senere i forløbet med at udføre interviewet, så gruppen havde mere konkret viden om emnet, og om hvilke spørgsmål, gruppen ville spørge professionelle på emnet om.

8.3 Programmet

Programmet er lavet som beskrevet i afsnittet "Implementering". Gruppen havde fra begyndelsen af løsningsdelen, valgt at lave løsningen i fugleflugtslinje. Dette er dog ikke det mest optimale, da der kan forekomme problemer i forhold til vejnettet i Aalborg, samt at der kan være bygninger eller andre forhold, der ikke medregnes, når der bliver målt i fugleflugtslinje. Dette har dog simplificeret programmet, i forhold til hvordan det laves, men det har også givet en usikkerhed, når den korteste rute skal bestemmes. Gruppen kan ikke garantere, at den givne rute i realiteten er den korteste, når der også skal tages højde for hvilke veje der rent faktisk kan benyttes.

Hvis gruppen skulle have implementeret en løsning, der tager højde for vejnettet, havde gruppen tænkt på to forskellige løsninger. Den første ville være at sætte hele Aalborg op i et grid, hvor vejnettet ville blive markeret med 1 og resten ville være markeret med 0. På den måde ville vejen kunne findes med forskellige søgealgoritmer, som fx A*. En anden løsning ville være at have en tabel, der indikerer hvilke veje der er forbundet og distancen der mellem. Disse løsninger ville have gjort ruten mere præcis, da den reelle korteste rute ville kunne findes.

For at beregne den korteste rute mellem attraktionerne, valgte vi i gruppen at benytte Nearest neighbour algoritmen, på grund af dens hurtige eksekveringstid, der gør det muligt for brugeren af programmet, at indtaste et højt antal attraktioner, uden at det går mærkbart ud over oplevelsen

med programmet. Algoritmen blev også valgt, på grund af den forholdsvis simple implementering af den, og at den generelt passede godt til vores behov. Problemet ved at bruge denne algoritme, er at den ikke nødvendigvis finder den korteste rute. Algoritmen er upræcis, og en fejlmargen bliver nødt til at accepteres, hvis ikke man vælger at gribe ind overfor algoritmen, hvis den er ved at gøre noget der tydeligt giver en længere rute end nødvendigt. Den korteste rute burde fx aldrig krydse sig selv, og det var muligvis en af de ting vi kunne have tjekket for, når algoritmen benyttes, for at sikre os at den i det mindste ikke gør det, og på den måde får en kortere rute, end hvis vi bare havde ladet den gennemføre sine beregninger.

En anden løsning, ville være at prøve alle ruter der overhovedet er for de valgte attraktioner, men eksekveringstiden stiger faktorielt med antallet af attraktioner, så der skal ikke vælges mere end et par stykker, før brugeren af programmet begynder at kunne mærke at det tager lang tid at lave beregningerne. Et scenarie vi helst gerne ville undgå. Double Minimum Spanning Tree og Dijkstra's var også algoritmer gruppen havde undersøgt, og prøvet at implementere, men da fokuset ikke var på at lave den korteste rute, men derimod give en interessant rute, blev simplicitet og hurtig eksekveringstid prioriteret. Derfor endte vi med at bruge Nearest neighbour algoritmen.

8.4 Den interessante rute

I dette projekt bliver den interessante rute skabt af selve brugeren. Dette sker når brugeren har valgt de attraktioner vedkommende ønsker at besøge. Derefter bliver brugeren spurgt om yderligere attraktioner, der ligger tæt på enten de valgte attraktioner eller ruten i mellem attraktionerne. I dette projekt vil det udgøre den interessante rute for brugeren selv.

En anden løsning ville være at gruppen dikterede en interessant rute for bruger. Dette skulle ske i form af brugeren skulle føres ned ad nogle zoner, fx gågaden og havnefronten, som gruppen har sagt vil være interessante.

Dette er to forskellige metoder at finde den interessante rute på, gruppen vælger dog ikke at diktere hvad den interessante rute er. Grunden til dette er, at den rute som gruppen diktere ikke nødvendigvis er den interessante rute for hver bruger af programmet, derfor valgte gruppen, at brugeren selv skulle tilføje ekstra attraktioner til sin rute. Gruppen ville dog godt have en blanding mellem disse to løsninger. Brugeren skulle selv bestemme hvad den interessante rute er for dem, de skulle dog blive adspurgt om byens mindre seværdigheder, der ikke tager lang tid mellem deres valgte attraktioner. Dette kunne være ting som gågaden, parker, havnefronten, statuer og lignende, så de ikke vil blive adspurgt om at tilføje tidskrævende attraktioner.

Konklusion & Perspektivering 9

I dette kapitel vil der blive konkluderet på, om løsningen besvarer den opstillede problemformulering fra problembeskrivelsen.

Problemformulering:

"Hvordan udvikles der en softwareløsning, der hjælper turisten med at finde rundt i en storby, på en interessant rute mellem turistens egne valgte attraktioner?"

Igen gennem processen af problemløsningen, blev der udviklet et stykke software, der hjælper turisten med at finde en interessant rute, i fugleflugtslinje. Denne fugleflugtslinje er der ikke et kort over, hvilket betyder, at turisten ikke bliver hjulpet i et særlig stort omfang. Hjælpen fra denne softwareløsning er blot et forslag for, hvilke attraktioner der skal besøges, og i hvilken rækkefølge med kortest mulig distance mellem disse. Den interessante rute udgøres af de attraktioner brugeren vælger, samt de attraktioner der er mulighed for at tilføje, når de bliver spurgt om yderligere attraktioner til deres rute, hvis der findes attraktioner tæt på deres nuværende rute. Et optimalt hjælpemiddel til at finde rundt i en storby ville være en løsning der kortlægger ruten, hvilket dette program ikke har formået. En eventuel løsning på dette, ville være at sætte attraktionerne ind på et kort, ved hjælp af de allerede benyttede koordinater. Problemet er at finde et kort der fungerer i C, og ikke fx JavaScript som Google Maps benytter.

Der kan tilnærmelsesvis siges, at det software der i denne rapport er udviklet stadig hjælper turisten med at finde rundt, da den giver en liste med rækkefølgen over attraktionerne, som brugeren har bestemt. Denne løsning kan ikke vise vej uden at brugeren selv indtaster attraktionerne i fx Google Maps. Programmet kan hverken kortlægge eller guide, men fortæller udelukkende om rutens planlagte rækkefølge.

Perspektivering

En fremmed storby kan være svær at finde rundt i, og det kan være svært at finde ud af hvilken rute er bedst at tage for at få set så meget som muligt, uden at spille tiden med at gå rundt og fare vildt. En softwareløsning kan være løsning på problemet, men den skal være smart, og fungere på mobile enheder, så brugeren altid kan trække den frem af lommen og få rutevejledning med det samme. Det kunne være det næste skridt for den softwareløsning der af gruppen er blevet udviklet, i hvert fald når den kan tage højde for veje, og ikke længere kun fungere i fugleflugt. Den ideelle løsning, der tidligere i rapporten er beskrevet, ville være der hvor projektet skulle ledes hen, hvis der skulle udvikles videre på projektet. En smartphone app, der også fungerer offline, er en god løsning, der både kan hjælpe udenlandske turister uden internetforbindelse, og turister fra samme land.

Problemet med at finde rundt på en flerpunktsrute er ikke begrænset til turister, der gerne vil se så mange attraktioner som muligt på en dag, men er også et problem hos fx hjemmeplejen,

varelevering, postbude eller lignende. Gruppens softwareløsning er derfor ikke begrænset til at løse et problem for turister, men kan i ligeså stor udstrækning løse det samme problem for andre brancher eller personer, selvfølgelig uden den del med den interessante rute. Forskellen er behovet for at få så præcis en rute som muligt, og hvor lang tid der kan ofres på at beregne ruten. Nearest neighbour algoritmen fungerer godt i tilfældet med turister, fordi det er en algoritme der kan beregne mange punkter hurtigt, men desværre også har sine usikkerheder, fordi den ikke nødvendigvis finder den korteste rute, men kun tilnærmelsesvist. Dette betyder ikke så meget for en turist der bare skal gå et par kilometer, men kan hurtigt komme til at betyde meget for firmaer der kører flere hundrede kilometer dagligt, og andre algoritmer vil derfor være nødvendige i andre sammenhæng.

Litteratur

- Center for Erhverv & Udvikling, 2012.** Center for Erhverv & Udvikling. *Erhverv og turisme*. <http://www.faxe-kommune.dk/erhverv-turisme>, 2012. Set d. 20/11-2014.
- Danmarks Statistik, 2008.** Danmarks Statistik. *Turismen - Regionalt, nationalt og internationalt*. <http://www.dst.dk/pukora/epub/upload/11676/tur08.pdf>, 2008. Set d. 19/11-2014 - side 8.
- Erhvervs- og Vækstministeriet, 2014.** Erhvervs- og Vækstministeriet. *Turisme*. <http://www.evm.dk/arbejdsomraader/internationalt-udsyn/turisme>, 2014. Set d. 16/12-2014.
- FindTheBestRoute.com, 2014.** FindTheBestRoute.com. *Route Optimization - Find the best route between multiple addresses*. <http://findthebestroute.com/RouteFinder.html>, 2014. Set d. 27/10-2014.
- Google, 2014.** Google. *Google Maps JavaScript API v3*. <https://developers.google.com/maps/documentation/javascript/directions>, 2014. Set d. 27/10-2014.
- Harboe, 2011.** Thomas Harboe. *Metode og projektskrivning*. 2011.
- Humerfelt, 1999.** Sigurd Hummerfelt. *Earth according to WGS 84*. http://home.online.no/~sigurdhu/Grid_1deg.htm, 1999. Set d. 11/12-2014.
- MatematikFessor, 2014.** MatematikFessor. *Projektion af Vektorer*. <https://www.matematikfessor.dk/lessons/projektion-af-vektorer-199>, 2014. Set d. 16/12-2014.
- Mathispower4u, 2013.** Mathispower4u. *Graph Theory: Dijkstra's Algorithm*. <https://www.youtube.com/watch?v=KvRwplnIoEM>, 2013. Set d. 09/12-2014.
- Nilsson, 2003.** Christian Nilsson. *Heuristics for the Traveling Salesman Problem*. <http://web.tuke.sk/fei-cit/butka/hop/htsp.pdf><https://www.matematikfessor.dk/lessons/projektion-af-vektorer-199>, 2003. Set d. 16/12-2014.
- Postamenter, 2014.** Postamenter. *Jordkloden*. <http://www.postamenter.dk/kloden/kloden.pdf>, 2014. Set d. 11/12-2014.
- Ritzau, 2014.** Ritzau. *Forbrugerrådet: Pengeløse butikker udelukker kunder*. <http://www.information.dk/telegram/496765>, 2014. Set d. 20/11-2014.
- Rosen, 2011.** Kenneth H. Rosen. *Discrete Mathematics and its Applications*. pdfbook. McGraw-Hill, 2011. s. 715.

TDC, 2014. TDC. *Data i udlandet.*

http://kundeservice.tdc.dk/privat/publish.php?dogtag=p_help_mob_ud_pdk_du, 2014. Set d. 18/11-2014.

VisitAalborg, 2014. VisitAalborg. *Sammen er vi både attraktive og stærke!*

<http://www.visitaalborg.dk/sites/default/files/asp/visitaalborg/uploads/partner2015/partner2015.pdf>, 2014. Set d. 20/11-2014.

VisitDanmark, 2013. VisitDanmark. *Fakta og tal om turismen i Danmark.*

<http://www.visitdenmark.dk/da/denmark/fakta-og-tal-om-turismen-i-danmark>, 2013. Set d. 26/10-2014.

Vitcenda, 2013. Mary Vitcenda. *Why should visitors love your town? Count the ways.*

<http://www.extension.umn.edu/community/news/tourism-assets.html>, 2013. Set d. 20/11-2014.

Vodafone, 2014. Vodafone. *Turismen - Regionalt, nationalt og internationalt.*

<http://www.vodafone.co.uk/shop/pay-monthly/travelling-abroad/vodafone-data-traveller/index.htm>, 2014.

Youth Central, 2014. Youth Central. *Planning your trip.* <https://www.youthcentral.vic.gov.au/travel-transport/travelling-overseas/planning-your-trip>, 2014. Set d.

27/10-2014.

Spørgeskema A

I dette appendiks vil teorien til spørgeskemaer blive beskrevet, og rådataen fra respondenterne præsenteret. Teorien er skrevet ud fra pdf'en om dataindsamling, som kan ses under [Harboe, 2011].

A.1 Teori

Spørgeskemaer er en form for interview, hvor den kvantitative metode benyttes, som kan ske både online og offline. I dette projekt er den online løsning blevet benyttet, hvilket kaldes et internetinterview. Den offline metode kaldes for postale undersøgelser. Denne metode foregår ved at respondenterne får spørgeskemaet fysisk i papirformat. Fordelen ved internetinterview er den hurtige respons fra respondenterne og den gør resultatbehandlingen nemmere. [1]

Når et spørgeskema skal udføres er det vigtigt at have helt styr på formålet, altså hvad er succes kriterierne, hvilke problemstillinger skal der være svaret på, efter spørgeskemaet er fuldført? Den anden ting som er vigtig at få afklaret, er hvem målgruppen for undersøgelsen er, hvem er det der skal svare på disse spørgsmål? Med disse to ting i baghovedet skal et spørgeskema så udføres, så formålet bliver opfyldt så godt som muligt, uden at gøre spørgeskemaet forvirrende for målgruppen.

A.1.1 Spørgsmålene

Spørgeskemaets spørgsmål skal først og fremmest være dækkende, på den måde at den skal kunne dække alle problemstillingerne som er blevet udformet. Derudover skal disse spørgsmål ikke gå ud over de opstillede problemstillingerne. Et spørgsmål som ikke svarer på en del af problemstillingen, er ikke brugbar i den efterfølgende analyse. Spørgsmålene skal være lige til sagen, der er ingen grund til for mange omveje. Som hovedregel er korte spørgsmål bedre end lange, da disse er mere direkte og overskuelige.

A.1.2 Formulering

Det er vigtigt i forbindelse med formuleringen af spørgsmålene til spørgeskemaet, at alle spørgsmålene vil blive forstået ens. Der må altså ikke være nogen tvivl hos respondenterne. Det er i denne forbindelse vigtigt ikke at bruge vage formuleringer, dobbeltspørgsmål, indforstået/faglig jargon og lange ord. Spørgsmålene kan give forskellige meninger for forskellige respondenter hvis disse ikke bliver overholdt, og dermed kan svarene ikke bruges til ret meget analytisk. Til sidst er ledende spørgsmål også farlige, da spørgeskemaundersøgelser som oftest har som mål at give objektive ikke forvrængede resultater.

A.1.3 Rækkefølge

Når rækkefølgen skal overvejes er der 4 ting der skal lægges særlig meget vægt på: -Motivationen hos respondenterne Respondenten vil efterhånden miste koncentrationen og motivationen til at besvare spørgeskemaet jo længere han eller hun kommer. Dette kan undgås ved først og fremmest kun at stille de nødvendige spørgsmål så spørgeskemaet bliver så kort som muligt. Derudover kan det være en fordel at stille de nemme spørgsmål i starten af spørgeskemaet så respondenterne får besvaret en masse spørgsmål i en fart, og derved kommer godt i gang. Det kan dog også være en god ide i nogle tilfælde at komme til sagen med det samme. -Konteksten spørgsmålet er indenfor I nogle tilfælde kræves der information fra et tidligere spørgsmål, før der kan svares på et andet. Derfor er det vigtigt at sørge for at rækkefølgen er således at respondenterne har fået stillet de krævede spørgsmål før det spørgsmål hvor respondenterne skal bruge informationen. -Den emnemæssige sammenhæng I rækkefølgen på spørgeskemaet skal der undgås alt for store emnemæssige brud, der skal være et naturligt flow mellem spørgsmålene. -Tragtmodellen Det er til stor fordel at stille de generelle spørgsmål før de specifikke, så der opstår en form for tragtmodel ned gennem spørgeskemaet.

A.2 Rådata

Antal respondenter: 60

Hvad er vigtigt for dig på din storbyferie? Sæt gerne flere krydser

Svarvalg	Besvarelser
Opleve kulturen	68.33% - 41
Se byens seværdigheder	90% - 54
Shopping	50% - 30
Fest/Bytur	26.67% - 16
Museumbesøg	11.67% - 7
Maden	63.33% - 38
Teater/Musik	18.33% - 11

Kommentarer: Sportsbegivenheder

I hvilken grad bliver din storbyferie planlagt?

Svarvalg	Besvarelser
Alt er planlagt til punkt og prikke inden ferie	0% - 0
Nogle ting er planlagt inden ferien	70% - 42
Planlægger dag for dag på ferien	28.33% - 17
Planlægger ikke	1.67% 1.67 - 1

Hvilke hjælpemidler bruger du til at planlægge din storbyferie?

- Tripadvisor
- Google
- Internettet
- Guide

- Google Maps
- Yelp
- Turen går til..
- Venner/familie som har været på stedet
- Rejsebureauer
- Bøger
- Lonely planet
- fdm-travel.dk
- booking.com
- hotels.com
- Momondo
- Ansrejser
- Expedia
- Hertz biludlejning
- Scout
- Lokale
- Kort
- Politikens de røde
- Top 10 vigtigste ting at se

Et program/applikation, som hjælper mig med at finde den hurtigste og/eller mest interessante vej igennem byen, via mine valgte "must see" destinationer, ville være noget jeg kunne bruge?

Hvilke redskaber bruger du til at finde rundt når du er på storbyferie? Vælg gerne flere

Svarvalg	Besvarelser
De lokale	46.67% - 28
Diverse kort/Brochure	81.67% - 49
GPS/Elektronisk kort	55% - 33
Taxa	10% - 6
Guider	18.33% - 11

Har du nogensinde haft problemer med at finde vej på din storbyferie?

Svarvalg	Besvarelser
Ja	68.33% - 41
Nej	31.67% - 19

Når du skal fra en aktivitet til en anden på din storbyferie, vil du helst tage den hurtigste rute eller en langsommere men mere interessant rute?

Svarvalg	Besvarelser
Den hurtige rute	20% - 12
Den interessante rute	80% - 48

Et program/applikation, som hjælper mig med at finde den hurtigste og/eller mest interessante vej igennem byen, via brugerens valgte "must see" destinationer, ville være noget jeg kunne bruge?

Meget uenig	Uenig	Enig	Meget enig
1.67% - 1	6.67% - 4	45% - 27	46.67% - 28

Interview B

B.1 Teori

Der er mange former for interviews og disse kan udføres på forskellige måder, men typisk når der snakkes om interviews, bliver de delt ind i tre forskellige former, enkeltinterview, spørgeskemaer og telefoniske interviews. Dette afsnit er udarbejdet ud fra PDF'en om dataindsamling, der kan ses i litteraturlisten, under [Harboe, 2011].

B.1.1 Enkeltinterview

Det enkelte interview forgår på følgende måde: Både interviewer og respondent mødes ansigt til ansigt. Fordelene herpå er tydelige, at give respondenter mulighed for at besvare private og intime spørgsmål, som en almen respondent ikke er tryk ved at tale om foran andre. Dette mindsker også chancen for at spørgsmålene bliver misforstået, samtidig med at svarene kan diskuteres på et højere plan, end ved et interview over telefonen eller ved et spørgeskema, da snakkes med mere end bare ord, nemlig kropssprog. Hvis interviewer har en god situationsfornemmelse kan et vellykket interview, forventes. En udvidelse af enkeltinterviewet kan der snakke om gruppeinterview. Metoden bruges hvis der er pres på tid og ressourcer. Metoden er den samme udover den forskel at der er flere respondenter. Der opfordres ikke til dialog mellem respondenterne. Et modsvar til denne metode er fokusgrupper. Denne metode opfordrer netop til dialog mellem respondenterne, men emnet her er temmelig afgrænset. Denne metode bliver brugt til at sammenligne skabelsen af holdninger i sociale miljøer og hvilke argumenter der bliver taget i brug.

B.1.2 Telefoninterview

Det telefoniske interview er lidt en sammenblanding af de to ovennævnte interview former. Telefoninterviewet foregår ved at en eller flere interviewere sidder bag røret og stiller en række spørgsmål, som på forhånd er fastlagte. Den væsentlige forskel på telefoninterviewet som er et kvalitativt interview og spørgeskemaet som er et kvantitativt interview, er at interviewerne kan uddybe deres spørgsmål på et højere plan, end et spørgeskema vil kunne. Måden hvorpå denne form for interview foregår er ved at scanne spørgeguiden ind i et program, hvorefter dette vil blive sendt til respondenter og unødvendige spørgsmål undgås. Her har respondenter så mulighed for at skrive sine egne svar ind, hvilket mindsker fejl ved fx transskription. Til sidst har interviewer mulighed for at gå i detaljer med hvert spørgsmål sammen med respondenter.[?]

B.1.3 Spørgeteknikker og metoder til interview

Når der snakkes om videnskabelige spørgeteknikker er det vigtigt at kende forskellene på dette og dagligdagssproget, som normalt bliver snakket. Der er de standardiserede spørgeteknikker, hvilket er hvor spørgsmålene og rækkefølgen på disse, omhyggeligt er blevet arbejdet med, og deres rækkefølge, er valgt på forhånd for interviewet. Denne metode er nyttig at tage i brug, hvis en interviewer kender problemstillingen. Her får intervieweren svar på sine spørgsmål med så lidt spildt information som muligt. Dette udføres typisk med spørgeskemaer. Nogle forskere mener, at et standardiseret interview også har det element, at forholdene og endda tiden for interviewet er ens for alle respondenter. Alle andre former for interview er indenfor kategorien ikke-standardiseret interview. Derudover er der de strukturerede interviews. Dette forgår lidt på samme måde, som de standardiserede interviews. Den væsentligste forskel herpå, er at spørgsmålene ikke er fastlagte, så det kun er spørgeguiden, der er fastsat. Dette skaber større mulighed for en kvalitativ interviewform, hvor intervieweren kan følge op på emner der kommer, som intervieweren ikke havde regnet med. Som et modsvar på denne form for interview, findes det ikke-strukturerede interview. Denne metode har hverken fastlagte spørgsmål, eller en fastlagt spørgeguide/rækkefølge på spørgsmål. Ved brug af denne metode kan intervieweren frit følge et givent emne ud fra respondentens svar, derfor kaldenavnet "det fleksible interview".

B.1.4 Lukkede og åbne spørgsmål

Lukkede spørgsmål bruges typisk ved kvantitative spørgeteknikker såsom et spørgeskema. Altså teknikker som gør at responsen let kan sammenlignes og analyseres. Denne metode af spørgsmål falder altså ind under kategorien standardiseret spørgsmål, da respondenterne hverken kan ændre på rækkefølgen af spørgsmålene eller gå ind og uddybe sine svar. Til hvert et træk, er der et modtræk. De åbne spørgsmål, som bliver benyttet i de kvalitative aspekter indenfor spørgeskemaer, altså den mulighed at respondenterne kan uddybe nogle svar, hvis intervieweren føler det er nødvendigt og stille sådan en plads til rådighed i spørgeskemaet. [Harboe, 2011]

B.1.5 Passive teknikker

Denne teknik går ud på at stille et spørgsmål, lade respondenterne svare, hvorefter intervieweren kommer ind med nogle spørgende kommentar. Ved brug af denne teknik, mindskes interviewernes bestemmelse i retningen af interviewet, og respondenterne kan komme med mere information, om et givent emne, og endda indbringe egne meninger og holdninger, hvis dette er vigtigt ift. emnet. [Harboe, 2011]

B.1.6 Aktiv spørgeteknik

Hvis intervieweren kommer ud for, at respondenterne er meget sky og tilbageholden med information, kan intervieweren manipulere ham/hende til at tro at den information de kommer ud med, er en lille del af en manglende kæde. Fx kan intervieweren nævne en given situation, hvorefter intervieweren spørger ind til det manglende led, altså den information intervieweren mangler. På denne måde ligner respondentens svar "bare" et lille manglende led i noget intervieweren allerede ved. [Harboe, 2011]

Interviewguide



- Præsentation af vores emne: Rutevejledning for turister i Aalborg
- Hvordan er turismen i Aalborg?
 - Hvor mange turister kommer der i gennemsnit pr. år i Aalborg (evt. bare 2013)?
 - Hvad kommer turisterne for at se/opleve i Aalborg?
 - Hvor mange turister kommer til jer, for at få hjælp?
 - Hvad hjælper i turisterne med, og hvordan hjælper i dem?
 - * Giver i dem ruter? Fortæller dem hvad er populært at se?
- Hvilke eksisterende løsninger findes der angående ruteplanlægning for turister?
 - Evt. andre ting til at hjælpe turisterne?
- Fortæl om vores idé: Ruten mellem attraktioner(hurtig/interessant), Fremtidige idéer fx om bruger rankings er en god idé, det med at man vælger sig ind på attraktioner via appen, hvor man også kan læse om attraktionerne.
- Fortælle at i vores spørgeskema at de gerne vil have den interessante rute. Hvad mener han kan være den interessante rute mellem attraktionerne?
- Hvad synes han om vores idé? Evt tilføjelser der kunne hjælpe os ?

Transskribering D

Mark: Vi fik et oplæg om at man kan bruge Google Maps til at finde en rute fra et punkt til et andet, men man har ikke muligheden for at angive mange punkter og få den bedste rute. Så det vi egentlig har fået til opgave, er at lave et sådan produkt, der kan det, og i den anledning tænkte vi at turister kunne værre en case. Sådan noget med at finde rundt mellem seværdigheder, finde ud af hvad de vil se, og så give dem en rute der kan vise dem rundt i Aalborg, eller enhver storby, men nu var det lige Aalborg vi havde fokus på. Så det er egentlig meget generelt det. Så har vi haft nogle spørgeskemaundersøgelser, hvor vi har spurgt folk om hvad de gerne vil se, og om de vil have fokus på at se det mest interessante ting i Aalborg, eller bare vil have en hurtig tur rundt. De svarede så, at de gerne ville have den mest interessante rute.

Mikael: Hvis de vidste at de skulle fra et punkt til et andet, eller den ene attraktion til den anden, om de så vil have nogle ideer til hvad der er den mest interessante vej, altså måske skulle de gå en km ekstra, men så fik de set nogle andre små attraktioner, eller om de bare ville have den hurtigste rute, hvor vi kom frem til at folk gerne vil have den mest interessante rute, hvor de kan krydse af hvad de vil ud og se.

Mark: Ideen for os, ville så være at lave en lille app, man har på sin telefon, hvor man så kan se at folk fx har ranked Aalborg Tårnet til at være 4/5, "har du så lyst til at se den" hvis folk kommer forbi. Det kunne værre sådan noget i den stil. Ja, så det vi først gerne vil høre noget om, er turisme i Aalborg.

Lars: Til alt det specifikke der, der henter jeg lige min kollega Kim, hvis han lige har tid, for der er jeg fuldstændig blank på viden om hvad vi lige gør, så lad os tage de der generelle spørgsmål først, og når vi så bliver lidt mere specifikke, så finder jeg lige ud af hvilken ven jeg skal ringe til.

Mark: Godt, jamen turisme i Aalborg. Hvordan, hvorledes, hvor mange og hvad vil de gerne se, og hvordan hjælper i dem rundt?

Lars: Altså, vi måler det jo på flere forskellige måder. Den mest anvendte, det er at se hvor mange overnatninger vi har i byen. Det ved vi jo fra byens hoteller, som indberetter det til Horesta, som så gør det tilgængelig hos Danmark Statistik, så vi kan gå tilbage og se hvor mange overnatninger der har været i juni måned 2014, og hvordan så det ud i forhold til juni måned 2013, og hvordan var fordeling på nationaliteter, var der fremgang på norske, var der tilbagegang på svenske, hvad med engelske, kinesiske osv. Osv. Så det er den måde vi normalt pejler efter, når vi snakker om hvorvidt det går godt eller skidt for turisme i Aalborg.

Men der er så lidt en ubekendt faktor, som vi ikke har nogle præcise tal på. Det er alle de gæster som kommer uden at overnatte. Det er fx de gæster der tager færgen oppe fra Kristiansand til Hirtshals og styre deres automobil til Bilka, hvor de voldboller supermarkedet derinde, fylder bagagerummet op med kølevare og hvad de eller lige har brug og behov for, og som i øvrigt er møg hamrende billigt, for i Norge koster alt 2-3 gange så meget som det gør i Danmark, og så går de typisk jo ind i midtbyen, eller i en af restauranterne derude, shopper lidt rundt og hygger sig lidt, inden de så om aftenen kører tilbage igen og tager færgen hjem. Dem har vi jo ikke så forfærdelig mange tal på. Vi ved hvor mange endagsturister Color Line har med færgen, men vi

ved rent faktisk ikke om de kører til Hjørring, eller kører til Aalborg eller til Århus. Vi ved af erfaring og fornemmelse, at rigtig mange kører til Aalborg fordi en af de primære grunde til at de kører hertil, er Bilka. Og sådan er der selvfølgelig også nogle andre målgrupper, hvor man kan sige der er nogle der falder uden for den der målbare ramme. Og det har vi ikke rigtig fundet ud af at få sat i system endnu. Hvor mange turister er der? Jamen jeg tror lige jeg vil tage at vise jer nogle nøgletal.

Lars går ud og henter nogle brochurer: Se nu får i lige en lille gave. Se det der er sådan set vores partnerkoncept. Det er det materiale vi bruger til at gå ud og erhvervet til at forstå at vi ligesom står sammen om den her markedsføring. VisitAalborg er jo sådan delvist kommunalt ejet. Det er kommunen der betaler den største del af vores drift. Lige knap 40% af de knap 15 millioner vi har i årlig driftsbudget, de er finansieret af kommunen. Resten, dem skaffer vi sådan set selv, ved hjælp af partnerskaber med erhvervet, hoteller, restauranter, attraktioner, detailhandelskæder, virksomheder, lufthavn, kongres og kulturcenter etc. etc. og desuden så gør vi også det at vi søger samarbejde både i regionen, med andre aktor uden for Aalborg, om at pulje pengene for at få så meget at skyde med som overhovedet muligt, men vi har også folk der sidder og arbejder på og lave projekt ansøgninger, forslag til de forskellige ministerielle og EU relaterede puljer, som kan give medfinansiering til nogle spændende og visionære projekter. Og det vi tilbyder ? det er sådan set delt op i et partnerkoncept som dels er basispartnerskab så er der nogle lidt dyrere business partnerskaber og VIP-partnerskaber som betaler 75.000kr om året, jeg skal ikke trætte jer med hvad det er, man kan ren faktisk læse det her i hvis i synes det er interessant hvad det er vi tilbyder, men nøgletallene som vi hviler på når vi skal definere os selv og vigtigheden af det vi gør, de står her nedenunder. I 2012 som er de seneste tilgængelige tal på omsætningen, ja der generede Aalborg en omsætning på lig knap 3mia kr. I relation til turister. En halv mio. til overnatninger, heraf det sådan set meget markant viser os hvor stor markedet det er. Heraf havde vi 140.000, ikke gæster, men roomnights, altså værelser solgt til norske gæster. Dertil skal så lægges de her norske endagsturister, som der også er rigtige mange af. Så hvis forbindelsen mellem Hirtshals og Norge eller Frederikshavn og Norge, den lige pludselige bliver oversvømmet eller afbrudt, så ville det betyde massive butikke? I Aalborg + at serviceerhverv virkelige komme på røven. Det kan man jo også se fordi at turismen alene står bag 4000 fuldtidsjob baseret på 2012 til ikke os?? Så vi er store , vier stærke og VisitAalborg arbejder, synes vi selv, top professionelt og i øvrigt Danmarks næststørste turisme-fremmeorganisation, der er kun København, Wonderfulcopenhagen som er vores kolleger i København, der er kun dem der er større, vi er mange gange større end de er i Århus og i Odense og de byer som vi ellers skulle normalt sammenligne os med. Så da vi er rent faktisk også gjort noget for ligesom ikke bare fastholde, men også udvikle tiltrækningen af flere gæster, ja, og så er der nogle andre nøgleting som kan have sine relevans.

Mikael: Hvad med den der vækst der?

Lars: Ja.

Mikael: Er det, ved i hvad det sådan skyldes, eller er det kun i Aalborg eller for hele?

Lars: Det jo en vækst i turisme og omsætning.

Mikael: Ja.

Lars: Det jo en vækst i turisme og omsætning som i 2012 var på 2,9mia. ikke. Der har man så regnet ud at i 2011, der var den så 4,6% lavere, så det sådan set bare en omskrivning af det, men det her det er ligesom for at det skal gerne virke som en imponator ikke, så folk siger hold da op, det er en fin vækst der må der være nogen ting vi har gjort rigtig. Så det er i forhold til året før.

Mikael: Ved i hvad det skyldes?

Lars: Ja, vi har fået noget mere løn, nej, det ved vi ikke, Jeg ved det ikke i hvert fald. Jeg har

ikke dyrket tallene i nøjagtighed, men det skyldes jo selvfølgelig forskellige ting, ikke også, det sjældent at man sådan kan finde en grund til noget som helst.

Mikael: Men jeg tænkte, når du siger med Norge at man så hvis man nu har fået bedre transport eller bedre skib fx at der kom de der 2 timers ture?

Lars: 3 timers, ja.

Mikael: Ja, 3 timers.

Lars: Godt tænkt.

Mikael: Så ved man ligesom.

Lars: Ja, godt tænkt.

Mikael: Så stiger..?

Lars: Ja det ender med at vi bliver nød til at pulse.

Mikael: der må man sikkert afsted meget.

Lars: Ja, ja, det har du sku helt ret, du kender lidt til det også kan jeg høre.

Mikael: Ja, jeg har taget det tog mange, eller jeg har taget den færge mange gange.

Mark: Han er nordmand.

Lars: Er du det?

Mikael: Hmm..

Efter en lang snak om noget irrelevant, spørger Lars: Hvad var det egentlig for et spørgsmål du stillede?

Mark: Har i også sådan noget guidet ture?

Lars: Ja, det har vi. VisitAalborg er jo et kommunalt foretagende, så vi skal passe meget på med ikke at gå ud og lave, hvad skal man sige, konkurrenceforvridende virksomhed, så vi søsætter egentlig ikke selv guidede ture, men vi gør alt hvad vi kan for at understøtte folk som tager nogle initiativer i retning af at skabe nogen gæsteoplevelser og det næsten uanset i hvilken retning det er, vi deltager gerne både med råd og vejledning og nogle gange hjælper vi også hvis vi kan komme afsted med det med økonomi til nogen event der er gæsterelateret og det uanset om det er til Aalborg eller det til "Forstår ikke.. hvilket race?" eller hvad det nu måtte være, eller det på den anden side er til erhvervsturisme, altså møder, kongresser, konferencer etc. Og desuden så er vi jo også som en seriøs og professionel operatør, optaget af at der bliver lavet nogle nye tiltage, at der bliver udviklet nogle nye muligheder som kan være med til at forbedre gæsteoplevelsen og det er uanset om det er guidet cykelture, eller kørsel med åben top-bus eller hvad det nu måtte være, vi har alle sammen spillet et rolle for at få det etableret, problemet er at i modsætning til København som har et noget større fundament at stå på med hensyn til antal gæster i byen, så er det lidt vanskeligere at etablere i en by som Aalborg fordi vi lige måske er niveauet under, der hvor det er nemt at få det til at blive til en rigtig god forretning, men det så på den anden side er der hvor vi så gør en stor indsats for at være med til at skabe det bedst mulige fundament for at de projekter vi så hjælper på vej også for den bedste chance for at skabe deres eget forretningsgrundlag, så det går vi aktivt ind og hjælper i, men det er ikke os der står som afsender at alt det, det må vi for det første ikke og for det andet så kan vi ikke, men vi finder nogle lokale aktører og arbejder sammen med fx i tilfælde med åben top bus, der arbejder vi sammen med et lokalt busfirma som vi hjalp med at finansiere indkøbet af sådan en åben top-bus og så kører den så 3 gange om dagen de 8 sommerferie uger, det så i øvrigt uden for sæsonen brugt til nogle andre formål, så der kunne laves en business case der hang nogenlunde sammen.

Mark: Attraktioner i Aalborg, hvad er der interessant at se? Hvad er der mange der tager til?

Lars: Ja, det er jo forholdsvis enkelt hvis man bruger det traditionelle. Hvis man bruger det utraditionelle, så er Aalborgs største turist attraktion, den ligger ud i City Syd, det IKEA. De

har 1.6 mio. årlige besøgende, tror jeg det er, så det er en meget stor turist attraktion. Det bliver selvfølgelig sagt med stor glimt i øjet og kæmpe smil, fordi de bliver ikke i traditionelt forstand opfattet som turistattraktioner, men det så sagt, samtidig med man har lidt på fornemmelsen, at der er nogen familier der bruger det som en slags søndagsunderholdning, og så tager de ud og får "köttbullar" og cola ad libitum, og så kører de hjem igen, og så har de haft en underholdende dag der. Men ellers, hvis man skal gå tilbage til det traditionelle, så er der ingen tvivl om, at vores allerstørste lokale attraktion, det er Aalborg Zoo, som jo har tæt på en halv million gæster om året. Hvis vi kigger lidt udenfor det, så er det jo klart Faarup Sommerland, og hvis vi går lidt mere ned på det jordnære niveau, med under 100 tusinde besøgende, jamen så er det nogle af de lokale museer, Lindholm Høje muset, forsvarsmuset, kunstmuset, Utzon osv. det er de attraktioner, som skal bære udfordringen med at skabe gæsteoplevelser på det traditionelle områder, der hedder attraktioner.

Mark: Så Aalborg tårnet er faktisk slet ikke noget?

Lars: Jo, jo det er det! Der er intet i vejen med Aalborg tårnet, men det eneste Aalborg tårnet som sådan kan, det er at tilbyde folk at komme op i en elevator og så nyde udsigten, ikke også? Og det er der så mange andre steder man kan. Og så kan man så få en omgang pommes frites deroppe.

Mark: Det koster vel ikke noget at komme op, gør det?

Lars: Jo, man betaler noget for at komme derop, med mindre du er inviteret i et eller andet selskab, så betaler man ikke noget for det.

Mikael: Men hvad siger du så, er der andre steder man kan se, hvis nu man skal ud og se oversigten over byen, som fx heroppe, men er der andre specielle steder man måske kan se gratis, som også er sådan lidt specielle?

Lars: Jamen hvis I gider at bruge at bruge lidt tid, så er der en masse turistinformationer på vores side, visitaalborg.dk, som i øvrigt er Aalborgs allermest besøgte site. Først og fremmest rettet mod gæsterne, men vi har også på fornemmelsen, at der er mange lokale der bruger den, vi har jo langt over 600.000 årlige unikke besøg, og man tæller først når man har navigeret på sitet i over 15 minutter, så der er rigtig rigtig meget tryk på det site. Og det kan vi jo også mærke, og det er bl.a. derfor vores partnere godt vil være med, fordi i ved, at når man har købt et basis partnerskab, så har man stort set fået sit eget profilering på vores web. Og det er stærke sager, især hvis man har noget og gå med der henvender sig til folk der er gæster i vores by, overnatninger, restauranter, attraktioner, specielle shoppingsmuligheder osv.

Mark: Men det er lidt sjovt du nævner shoppingsmuligheder, for vi havde en diskussion i dag med vores vejleder, om hvorvidt shopping er en attraktion, for vi mente, at hvis man nu skulle fra fx Aalborg Zoo til Karolinelund, så kan man tage den hurtige rute, men det kunne være sjovt at komme forbi gågaden, hvor vi så tænkte, det er da lidt en attraktion, altså noget at opleve?

Lars: Jamen det er jeg da helt enig med jer i.

Mikael: Eller i hvert fald til den interessante rute.

Lars: Men jeg tror bare, at selvom jeg er en gammel mand, så vil jeg bare sige, at man skal begynde at tænke udenfor boksen, fordi, hvad er det der gør at folk rejser tilbage til Aalborg, eller tilbage til Barcelona, eller hvor det nu er i verden, ikke også. For jeg ved udmærket godt, at når jeg tager til Barcelona, så er det først og fremmest fordi der er en utrolig behagelig temperatur dernede, når vi er på den her tid af året, ikke også, så er det typisk 10-12 grader varmere typisk i Barcelona. Og når jeg går med efterårsjakke på her, kan jeg nøjes med t-shirt ned af "la ramblaen". Og det andet det er så det folkeliv der er på "ramblaen" osv, og der tror jeg, at noget af det der gør Aalborg mere attraktivt, det er sgu ikke, at de har fået en ny F16 fly, ude på museet eller en ny girafunge oppe i Zoo osv. Det er jo hele det liv som Aalborg er på vej til at udvikle,

og et stykke hen af vejen, har udviklet, med de nye åbne områder på havnen, hvor man skaber atmosfære, og oplevelsesrum, hvor folk gerne vil være. Derfor synes jeg også at vores gågader de hører med til oplevelsen, jeg tror ikke, at der er nogen der rejser fra Sydnorge, ja de er så lidt specielle, lad os hold Norge udenfor, men jeg tror ikke, at der er nogen der rejser fra Tyskland, eller England eller Amsterdam, hvor vi nu også har en forbindelse til, eller fra Frankfurt, som vi snart har en forbindelse til, for at handle net undertrøjer ind i H&M. Og jeg tror heller ikke, at de går ind for at købe en skjorte inde hos Torben Walder for så vidt, men jeg tror på, at de har lyst til at gå ud og finde nogle af de butikker, som fx kan levere noget Danish design, jeg tror på at de kan lide at gå ind i de butikker, hvor man kan levere noget unikt kunsthåndværk, ler-klaskeri, glaspusteri osv. som vi trodsalt har nogen forskellige muligheder for i Aalborg. Jeg tror de kan lide at sætte sig ned, de steder, hvor der er en behagelig fortovscafe/restaurant, for at få et lille glas og lidt at spise og nyde det liv der foregår rundt omkring, så jeg tror godt at vi kan sige, til city foreningen, for turristoplevelsens skyld, så er det ikke vigtigt, at H&M overlever med 3 butikker i gågade systemet i Aalborg, det er vigtigt, at Inspiration og at Georg Jensen Damask og nogle af de specielle butikker, der har noget specielt og unikt, at de overlever, fordi de er med til at sætte en streg under og skabe, måske ikke skabe en "reason to go", men i hvert fald være med til at gøre besøget i Aalborg til noget originalt og unikt. Hvis man køber et eller andet, som man er glad for, så kan man altid huske hvor man har købt det henne, ikke? Og hvis man nu har købt det i Aalborg, så er det jo en udmærket præference at have.

Mark: Jeg ved ikke om vi har meget mere?

Lars: Nej, men jeg vil gerne i relation med det her Google Maps osv. så vil jeg godt lige bede min kollega Kim om at komme ind.

Mark: Men lige nu er vi kun i gang med analyse, vi skal først i gang med at snakke om programmering sådan lidt senere, det bliver nok om en halv måneds tid, så måske lidt senere.

Mikael: Vi er kun i gang med at finde ud af, om det er relevant det her med ruteplanlægning.

Lars: Årh, cool.

Mikael: Det er egentlig bare det.

Lars: Men det kan jeg jo ikke svare jer på, men det kan Kim måske svare jer på, ved at sige det er en god vej at køre ud af, eller også sige, vi har allerede "bum bum bum", ikke?

Mark: Nå jo, ja, eksisterende løsninger, ikke? Hvad har i af sådan nogen?

Lars: Ja, men skal jeg ikke bare kalde på ham alligevel?

Mikael: Jo.

Lars: Det er så cool, og generelt har vi det sådan i Visit Aalborg, at alle der kommer med et projekt, dem vil vi gerne give de bedste redskaber til, at gå i den mest hensigtsmæssige retning. Og grunden til, at jeg synes det var relevant, at Kim også kom ind er fordi, at han ved lidt om, hvad der er, og hvad vi drømmer om osv. Der er jeg fuldstændig blank. Især når vi kommer over i en web-baseret virkelighed. Og samtidig med det, skal det ikke være en hemmelighed, at hvis ikke vi havde en god dialog med uddannelsesstederne, så ville vores kapacitet blive reduceret med adskillige procenter. Jeg ved ikke om det er 10% eller 15%, det er måske endda 20%. Men vi har mellem fire og seks praktikanter fast baseret i huset, og vi er typisk ude efter dem, der er mindst er ude i en tre til fire måneder af gangen, for også at få noget ud af investeringen, så de kan levere noget tilbage for alt den værdifulde information de får proppet ind i hovedet af os. Det skal man ikke undervurdere, at komme med til vores fredagsbarer. Så det synes jeg gerne vi vil bidrage til. Nu synes jeg i skal til at fortælle lidt om hvor i er henne, og det her er meget gryn osv, så take it away boys.

Mark: Vi læser Software ude på universitetet på Strandvejen, og vi skal skrive projekt om ruteplanlægning, hvor vores initierende problem er, at GoogleMaps kun kan vise en rute fra et

sted til et andet.

Kim: I vil gerne lave en form for flerpunktsrute?

Mark: Ja præcis, og hvordan man finder den bedste. Så har vi turisme som case, fordi det var interessant, hvad turisterne havde lyst til, og hvis de havde en masse seværdigheder de gerne ville se, hvordan kommer de så fra det ene sted til det andet. Og om de bare har lyst til at komme fra det ene sted til det andet, eller om de også har lyst til, hvis nu programmet kan foreslå en mere interessant rute, som andre har set eller rated højt. Lige nu er vi kun igang med at høre om turisme i Aalborg, fordi vi ikke er så langt med projektet endnu.

Kim: Men i stedet for at folk kommer til at tænke ud af et spor, og tænker "det har vi allerede lavet", eller et eller andet, så er det måske relevant lige at spore den en lille smule.

Mikael: Indtil videre har vi fundet TripAdvisor, som kan finde frem til et sted. Det er den eksisterende løsning der ligger tættest op af det, vi har tænkt os at lave.

Mark: Det var mere om i havde et eller andet?

Kim: Nej, vi har ikke noget ala det faktisk, så det lyder da rigtig spændende. Det lyder som en god idé i hvert fald. Det lyder interessant for både turisten og os. Så det ville blive rigtig godt at pushe videre til brugerne.

Mark: Der er ikke så meget i det, da vi ikke er så langt, kan man sige.

Kim: Vi kan da helt sikkert hjælpe jer med, at tilrettelægge lidt om, hvad er vores erfaring med, hvad turisterne gerne vil have. Tendensen med sådanne programmer er tit, synes jeg, at man prøver at lave et program der kan helt vild meget, og så ender en med at være værdiløs, da den kan alt for meget, og ikke har noget fokus. Så det der med at snævre det lidt ned og sige, "vores program, den kan lige præcis hjælpe dig med at finde den her tur". Det er det, der får en til at vælge at downloade den.

Mark: Vi har fået at vide, at vi kun må skrive i programmeringssproget "C", så vi må egentligt kun lave et lille program til en computer.

Lars: Der skal jo mange sten til at bygge et hus, så hvis man skaber et stærkt fundament, så kan man bygge videre på det på et tidspunkt. Så er der måske bedre plads i forløbet til det.

Mikael: Så har man måske også nogle bedre redskaber.

Kim: Det er vel en art af forundersøgelse af et man kan tage fat på senere hen, men vi vil da helt sikkert gerne hjælpe.

Lars: Men gutter, nu har i jo også Kim direkte at referere til. Det skal retfærdigvis siges til vores og til Kims og hele husets undskyldning, at lige nøjagtig vore afdeling er lidt hårdt spændt for i øjeblikket. Vi har vores web. Jeg så i øvrigt at i fredagsmailen, at vores webmaster er gået på barsel, så nu laver vi en side til nyfødte forældre. Nu er han jo på barsel, og vi har en praktikant deroppe.

Kim: Vi har tit noget i gang, og vi vil helt vild gerne. Fordi vi også får en masse ud af det, vi får noget viden og nogle skide gode idéer fra jer, og nogle gange kører det også videre, de projekter i nu har gang i. Så vi er altid klar på en indledende snak og hjælp med lidt retning på projektet.

Lars: Kim, tak for det. Nu vil jeg slutte af med en lille snak. Det hører også med til historien, at vi har i mange år udgivet forskellige publikationer, igen for at gøre hvad vi kan, for at skabe den bedst mulige gæste-oplevelse. Det har vi selvfølgelig dels gjort i "værgefasen", altså den fase når vi foreksempel tager til syd-norge eller nord-tyskland, eller til england eller whatever, hvor vi prøver at værgе gæster til vores by. Men også i hele gennemførelses-fasen, i den tid hvor folk er kommet, så sikrer at de får en god oplevelse. Så har vi blandt andet haft det her by-kort, Aalborgs eneste autoriseret by-kort, der er flere der har prøvet på at planke os, og lave noget tilsvarende. Men de når os ikke til sokke-holderne, for det første fordi, at vi er det eneste kort, der har turist informationer, og værdifulde oplysninger, og for det andet er vi også de eneste udgivere

som regelmæssigt distribuerer kortet ud til nøglesteder, hvor gæsterne er. Hoteller, restauranter, butikker, overnatningssteder, alle de steder hvor turisterne kommer fra, står de her steder før man kommer til Aalborg. Så det er delt sådan op, at vi har det store by-kort, inklusiv City-Syd osv. på den ene side, og Nørresundby, som er rigtig vigtig, på den anden side. Og på modsatte side af kortet, der har vi så det, som vi må sige er den mest anvendte, er vores indre by-kort. Her refererer vi til alle mulige forskellige institutioner, som har overnatnings ting osv. For godt to år siden, har vi skåret alle vores printmedier fuldstændig væk, fordi der synes vi, at vi var blevet elektroniske, og nu skulle alt ske på QR-kode basis osv. Men vi måtte erkende at vi havde gjort regning uden værk, det var for tidligt, folk har stadigvæk brug for, og vil gerne have, et stykke papir i hånden. Men så gik vi lidt i tænkeboks, og har nu sidenhen lavet nogle nye publikationer, som vi gjort første gang i år, hvor vi lavede en lille Quick-Guide. Man kan ikke annoncere i den, men man kan købe en optagelse, hvor man har en halv side. Det er så kun for virksomheder inden for shopping, dining og attraktioner. Altså restauranter, caféer og så attraktioner, det er kun dem vi vil have her. Fordi vi gerne vil have, at det skal være en slags guide, som folk kan bruge når de er kommet til byen og går rundt. Den bliver distribueret i 25.000 eksemplarer, og den har vi lavet 250.000 eksemplarer af. Altså $\frac{1}{4}$ million eksemplarer, den gælder så i et helt år. Og 20.000 af de 250.000 bliver lavet på engelsk. Den anden her er så lavet sådan, at der inde i midten er et lille mini-udprint af vores by-kort, så man kan finde rundt i byen, fordi det er formålet – Der er ikke nogen grund til at lede folk ud i City-Syd, da de fleste vil være inde i byen. Så er der lidt praktisk information, attraktioner, Zoo, Utzon centeret, Kunsten, Lindholm Høje, osv. Og så der er noget café-halløj, restauranter, og sidst men ikke mindst nogle attraktioner, som shopping-muligheder, som har valgt at købe en optagelse. Den gentager vi så i 2015, plus at vi laver, målrettet mod sommergæsterne, et magasin i A4-format, som også kommer i 25.000 eksemplarer, som først og fremmest skal lave lidt redaktionelt samt noget annoncering, teasing for, hvad man kan lave hen over sommeren i Aalborg. Hvor vi blant andet i år havde noget Tall-Ships-Race, og Blå Festival, og flere forskellige andre større projekter på Beddingen. Og som noget andet nyt, laver vi også dette magasin format som en cruise-udgave, som er målrettet mod de gæster der kommer fra sø-siden, vi er jo i den heldige situation, at vi skal være værter for 15 stor-krydstogtskibe, og næsten 15.000 gæster henover sommeren 2015. Vi tilbyder gæsterne en unik oplevelse, og lægger ekstremt meget vægt på værtsskab, og især det gode værtsskab. Måden at gøre det på, er ved at give dem oplevelser som er unikke, og som rammer dem i hjertet på en eller anden måde, og der valgte vi for sjov skyld, for et par år siden, at i stedet for at have et orkester som stod og spillede musik, så valgte vi at lave en give-away, så de fik en rigtig dansk hotdog, serveret fra en pølsevogn, og det synes folk var helt skørt. Men de elsker jo smagen, og de synes jo det er fantastisk. Det kan de jo ikke få nogen som helst andre steder på deres cruise. Det blev en rigtig populær begivenhed. Men gutter, jeg synes i skal gå tilbage til jeres studie-kammer, og så håber jeg at det har bragt lidt info osv.

Program koden



```
1 #include <stdio.h>
   #include <stdlib.h>
3  #include <string.h>
   #include <math.h>
5
   #define KM_PR_LNDGRAD 60.7722
7  #define KM_PR_BRDGRAD 111.36
   #define ANTAL_ATTRAKTIONER 22
9  #define ANTAL_KANTER ANTAL_ATTRAKTIONER*(ANTAL_ATTRAKTIONER-1)
   #define MAX_STRING 50
11
   typedef struct {
13     char navn[MAX_STRING];
       char adresse[MAX_STRING];
15     double kmFraGreenwich, kmFraAekvator;
       int besøegt;
17 } attraktion;

19 typedef struct {
       double ruteLaengde;
21     attraktion rute[ANTAL_ATTRAKTIONER];
   } naboRute;
23
   typedef struct {
25     attraktion start;
       attraktion slut;
27     double laengde;
   } kant;
29
   typedef struct {
31     double x, y;
   } vektor;
33

   void initialiserAttraktioner(attraktion *attraktioner);
35 void udregn_kanter(attraktion *attraktioner, kant *kanter);
   double beregn_dist(attraktion startAttraktion, attraktion slutAttraktion);
37 void valgafAttraktioner(attraktion *attraktioner, attraktion *valgteAttraktioner, int
       *antalValgteAttraktioner, attraktion *ikkeValgteAttraktioner);
   void findKortesteNaboRute(attraktion *valgteAttraktioner, int antalValgteAttraktioner, attraktion
       *ruteAttraktioner, kant *kanter, double *samletLaengde);
39 void findNaboRute(attraktion *valgteAttraktioner, int antalValgteAttraktioner, attraktion
       *startAttraktion, kant *kanter, attraktion **tempRute, double *ruteLaengde);
   double findDist(attraktion start, attraktion slut, kant *kanter);
41 void findEkstraAttraktioner(attraktion *ruteAttraktioner, int *antalValgteAttraktioner,
       kant *kanter, attraktion *ikkeValgteAttraktioner, double maxDist, attraktion
       *ekstraAttraktioner, int *antalEkstraAttraktioner);
43 void findEkstraAttraktionerFirkant(attraktion startAttraktion, attraktion slutAttraktion, kant *kanter,
       attraktion attraktionAtTilfoeje, double maxDist,
           attraktion *ekstraAttraktioner, int *antalEkstraAttraktioner);
45 double prikProdukt(vektor vektor1, vektor vektor2);
   void valgAfEkstraAttraktioner(attraktion *valgteAttraktioner, int *antalValgteAttraktioner, attraktion
       *ekstraAttraktioner, int antalEkstraAttraktioner);
47 void aendre_startsted(attraktion *ruten, attraktion nytStartSted, int antalAttraktioner, attraktion
       *outputRute);
   void outputTilFil(attraktion *ruteAttraktioner, int antalValgteAttraktioner);
```

```

49 void kopierFil(FILE *filSkabelon, FILE *filOutput);
   int attraktionErTilfoejet(attraktion *ekstraAttraktioner, int antalEkstraAttraktioner, attraktion
       attraktionAtTilfoeje);
51
52 int main()
53 {
54     attraktion attraktioner[ANTAL_ATTRAKTIONER];
55     attraktion valgteAttraktioner[ANTAL_ATTRAKTIONER];
56     attraktion ikkeValgteAttraktioner[ANTAL_ATTRAKTIONER];
57     attraktion ruteAttraktioner[ANTAL_ATTRAKTIONER+1];
58     attraktion endeligRute[ANTAL_ATTRAKTIONER+1];
59     attraktion ekstraAttraktioner[ANTAL_ATTRAKTIONER];
60     kant kanter[ANTAL_KANTER];
61     double samletLaengde = 0;
62     int antalValgteAttraktioner = 0, antalEkstraAttraktioner = 0;
63
64
65     initialiserAttraktioner(attraktioner);
66
67     udregn_kanter(attraktioner, kanter);
68
69     valgafAttraktioner(attraktioner, valgteAttraktioner, &antalValgteAttraktioner, ikkeValgteAttraktioner);
70
71     findKortesteNaboRute(valgteAttraktioner, antalValgteAttraktioner, ruteAttraktioner, kanter,
        &samletLaengde);
72
73     findEkstraAttraktioner(ruteAttraktioner, &antalValgteAttraktioner, kanter,
        ikkeValgteAttraktioner, 0.1, ekstraAttraktioner, &antalEkstraAttraktioner);
74
75     if(antalEkstraAttraktioner > 0){
76         valgAfEkstraAttraktioner(valgteAttraktioner, &antalValgteAttraktioner, ekstraAttraktioner,
            antalEkstraAttraktioner);
77
78         findKortesteNaboRute(valgteAttraktioner, antalValgteAttraktioner, ruteAttraktioner, kanter,
            &samletLaengde);
79     }
80
81     aendre_startsted(ruteAttraktioner, valgteAttraktioner[0], antalValgteAttraktioner+1, endeligRute);
82
83     outputTilFil(endeligRute, antalValgteAttraktioner);
84
85     if(antalValgteAttraktioner != 0){
86         int i;
87         printf("\nDin rute:\n");
88         for(i = 0; i < antalValgteAttraktioner+1; ++i){
89             printf("%s\n", endeligRute[i].navn);
90         }
91         printf("\nRutens samlede laengde: %.2lfkm\n", samletLaengde);
92     }
93     return 0;
94 }
95
96
97 void initialiserAttraktioner(attraktion *attraktioner){
98     FILE *input_file_pointer;
99     int i = 0;
100     double lndgrad;
101     double brdgrad;
102
103     input_file_pointer = fopen("attraktioner.txt", "r");
104
105     if(input_file_pointer != NULL){
106         while(fscanf(input_file_pointer, " %s %s %lf %lf", attraktioner[i].navn, attraktioner[i].adresse,
            &brdgrad, &lndgrad) == 4){
107             attraktioner[i].kmFraGreenwich = lndgrad * KM_PR_LNDGRAD;
108             attraktioner[i].kmFraAekvator = brdgrad * KM_PR_BRDGRAD;
109             attraktioner[i].besoegt = 0;
110             i++;

```



```

111     }
112     }else{
113         printf("kunne ikke aabne fil\n"); exit(1);
114     }
115     fclose(input_file_pointer);
116 }
117
118 void udregn_kanter(atrakktion *attraktioner, kant *kanter)
119 {
120     int i;
121     int y;
122     int indexTilKanter = 0;
123     for (i = 0; i < ANTAL_ATTRAKTIONER; i++)
124     {
125         for (y = i; y < ANTAL_ATTRAKTIONER; y++)
126         {
127             if (strcmp(attraktioner[i].navn, attraktioner[y].navn) != 0)
128             {
129                 kant k, j;
130                 k.start = attraktioner[i];
131                 j.slut = k.start;
132                 k.slut = attraktioner[y];
133                 j.start = k.slut;
134                 k.laengde = beregn_dist(k.start, k.slut);
135                 j.laengde = k.laengde;
136                 kanter[indexTilKanter] = k;
137                 kanter[indexTilKanter+1] = j;
138                 indexTilKanter += 2;
139             }
140         }
141     }
142 }
143
144 double beregn_dist(atrakktion startAttraktion, atrakktion slutAttraktion)
145 {
146     return sqrt(pow(startAttraktion.kmFraGreenwich - slutAttraktion.kmFraGreenwich, 2) +
147                 pow(startAttraktion.kmFraAekvator - slutAttraktion.kmFraAekvator, 2));
148 }
149
150 void valgafAttraktioner(atrakktion *attraktioner, atrakktion *valgteAttraktioner, int
151     *antalValgteAttraktioner, atrakktion *ikkeValgteAttraktioner){
152     int i = 0, j = 0, k = 0, l = 0, m = 0, n = 0, valgt = 0, y = 0;
153     int opretNy = 1;
154
155     printf("Rutevejledning - A401 - P1\n");
156     printf("Dette program vil, ud fra dine oenskede attraktioner, bestemme den korteste rute.\n");
157     printf("Du kan se de attraktioner du kan vaelge her:\n");
158
159     int halvdelen = ANTAL_ATTRAKTIONER/2;
160
161     for(i = 0; i < halvdelen; i++){
162         printf("%d: %-40s \t%d: %s\n", i+1, attraktioner[i].navn, i+halvdelen+1,
163             attraktioner[i+halvdelen].navn);
164     }
165
166     printf("Vaelg de attraktioner du oensker at se ved at skrive det tilhoerende tal.\n");
167     printf("Vaelg et tal (svarende til en attraktion) af gangen og tryk enter efter hver indtastet tal.\n");
168     printf("Tast 0, efterfulgt af enter, for at gaa videre. Indtast ikke samme tal 2 gange.\n");
169     do{
170         opretNy = 1;
171         if (scanf("%d", &k) == 1){
172             for (y = 0; y < j; y++)
173             {
174                 if (strcmp(valgteAttraktioner[y].navn, attraktioner[k-1].navn) == 0)
175                 {
176                     printf("Du har allerede indtastet denne attraktion. Proev igen.\n");
177                 }
178             }
179             if (k == 0)
180             {
181                 printf("Du har valgt ingen attraktioner. Proev igen.\n");
182             }
183             else
184             {
185                 for (i = 0; i < j; i++)
186                 {
187                     if (strcmp(attraktioner[i].navn, attraktioner[k-1].navn) == 0)
188                     {
189                         printf("Du har allerede valgt denne attraktion. Proev igen.\n");
190                     }
191                     else
192                     {
193                         attraktioner[i].valgt = 1;
194                     }
195                 }
196                 for (i = j; i < ANTAL_ATTRAKTIONER; i++)
197                 {
198                     attraktioner[i].valgt = 0;
199                 }
200                 j++;
201                 *antalValgteAttraktioner++;
202                 *ikkeValgteAttraktioner = attraktioner[j];
203                 opretNy = 0;
204                 valgt++;
205             }
206         }
207     } while (opretNy == 1);
208 }
209
210 void print_attraktioner(atrakktion *attraktioner)
211 {
212     for (i = 0; i < ANTAL_ATTRAKTIONER; i++)
213     {
214         printf("%d: %-40s \t%d: %s\n", i+1, attraktioner[i].navn, i+halvdelen+1,
215             attraktioner[i+halvdelen].navn);
216     }
217 }
218
219 void print_valgte_attraktioner(atrakktion *valgteAttraktioner, int antalValgteAttraktioner)
220 {
221     for (i = 0; i < antalValgteAttraktioner; i++)
222     {
223         printf("%d: %-40s \t%d: %s\n", i+1, valgteAttraktioner[i].navn, i+halvdelen+1,
224             valgteAttraktioner[i+halvdelen].navn);
225     }
226 }
227
228 void print_ikke_valgte_attraktioner(atrakktion *ikkeValgteAttraktioner, int antalIkkeValgteAttraktioner)
229 {
230     for (i = 0; i < antalIkkeValgteAttraktioner; i++)
231     {
232         printf("%d: %-40s \t%d: %s\n", i+1, ikkeValgteAttraktioner[i].navn, i+halvdelen+1,
233             ikkeValgteAttraktioner[i+halvdelen].navn);
234     }
235 }
236
237 void menu(atrakktion *attraktioner)
238 {
239     int i;
240     for (i = 0; i < ANTAL_ATTRAKTIONER; i++)
241     {
242         printf("%d: %-40s \t%d: %s\n", i+1, attraktioner[i].navn, i+halvdelen+1,
243             attraktioner[i+halvdelen].navn);
244     }
245 }
246
247 void main()
248 {
249     atrakktion *attraktioner;
250     atrakktion *valgteAttraktioner;
251     atrakktion *ikkeValgteAttraktioner;
252     int antalValgteAttraktioner;
253     int antalIkkeValgteAttraktioner;
254     menu(attraktioner);
255     valgafAttraktioner(attraktioner, valgteAttraktioner, &antalValgteAttraktioner,
256         ikkeValgteAttraktioner);
257     print_attraktioner(attraktioner);
258     print_valgte_attraktioner(valgteAttraktioner, antalValgteAttraktioner);
259     print_ikke_valgte_attraktioner(ikkeValgteAttraktioner, antalIkkeValgteAttraktioner);
260 }

```

```
        opretNy = 0;
177     }
    }
179     if (k == 0)
        break;
181     else if (k > ANTAL_ATTRAKTIONER || k < 0)
        printf("Tallet svarer ikke til en attraktion\n");
183     else if (opretNy && (k <= ANTAL_ATTRAKTIONER))
    {
185         valgteAttraktioner[j] = attraktioner[k-1];
        printf("Tilfoejet attraktion: %s\n", attraktioner[k-1].navn);
187         j++;
    }
189 }
else
191 {
    printf("Fejlindtastning - proev igen\n");
193     char e[MAX_STRING];
    scanf("%s", e);
195     k = 1;
}
197 } while(j < ANTAL_ATTRAKTIONER && k != 0);

199

201 for (l = 0; l < ANTAL_ATTRAKTIONER; ++l)
{
203     valgt = 0;
    for (m = 0; m < j; ++m)
205     {
        if(strcmp(attraktioner[l].navn, valgteAttraktioner[m].navn) == 0){
207             m = j;
            valgt = 1;
209         }
    }
211     if(valgt != 1){
        ikkeValgteAttraktioner[n] = attraktioner[l];
213         n++;
    }
215 }
*antalValgteAttraktioner = j;
217 }

219 void findKortesteNaboRute(attraktion *valgteAttraktioner, int antalValgteAttraktioner, attraktion
    *ruteAttraktioner, kant *kanter, double *samletLaengde){
    /* input er valgteAttraktioner arrayet, og kanter arrayet*/
221    /*output er ruteAttraktioner og samletLaengde*/
    double ruteLaengde;
223    attraktion *tempRute[antalValgteAttraktioner+1];
    *samletLaengde = 100000;

225
    int i;
227    int h;
    int j;
229    for (i = 0; i < antalValgteAttraktioner; ++i)
    {
231        for (h = 0; h < antalValgteAttraktioner; ++h)
        {
233            valgteAttraktioner[h].besoegt = 0;
        }
235        findNaboRute(valgteAttraktioner, antalValgteAttraktioner, &valgteAttraktioner[i], kanter, tempRute,
            &ruteLaengde);
        if(ruteLaengde < *samletLaengde){
237            *samletLaengde = ruteLaengde;
            for (j = 0; j < antalValgteAttraktioner+1; ++j)
239            {
                ruteAttraktioner[j] = *tempRute[j];
            }
        }
    }
}
```

```

241     }
242     }
243 }
244 }
245
246 void findNaboRute(atrakktion *valgteAttraktioner, int antalValgteAttraktioner, atrakktion
    *startAttraktion, kant *kanter, atrakktion **tempRute, double *ruteLaengde){
247     int i = 0;
248     double lavesteLaengde = 10000;
249     *ruteLaengde = 0;
250
251     tempRute[i] = startAttraktion;
252
253     for(i = 0; i < antalValgteAttraktioner-1; ++i){ /*da der f.eks. kun er 4 kanter imellem 5 punkter*/
254         tempRute[i]->besoegt = 1;
255         int j = 0;
256         for (j = 0; j < antalValgteAttraktioner; ++j)
257         {
258             if(valgteAttraktioner[j].besoegt != 1 && findDist(*tempRute[i], valgteAttraktioner[j], kanter) <
259                 lavesteLaengde){
260                 lavesteLaengde = findDist(*tempRute[i], valgteAttraktioner[j], kanter);
261                 tempRute[i+1] = &valgteAttraktioner[j];
262             }
263         }
264         *ruteLaengde += lavesteLaengde;
265         lavesteLaengde = 10000;
266     }
267     tempRute[antalValgteAttraktioner] = startAttraktion;
268     *ruteLaengde += findDist(*tempRute[antalValgteAttraktioner-1], *tempRute[antalValgteAttraktioner],
269         kanter);
270 }
271
272 double findDist(atrakktion start, atrakktion slut, kant *kanter){
273     int i;
274     for (i = 0; i < ANTAL_KANTER; ++i)
275     {
276         if(strcmp(kanter[i].start.navn, start.navn) == 0 && strcmp(kanter[i].slut.navn, slut.navn) == 0){
277             return kanter[i].laengde;
278         }
279     }
280     printf("Kunne ikke finde passende kant\n"); exit(0);
281 }
282
283 void findEkstraAttraktioner(atrakktion *ruteAttraktioner, int *antalValgteAttraktioner,
    kant *kanter, atrakktion *ikkeValgteAttraktioner, double maxDist, atrakktion
    *ekstraAttraktioner, int *antalEkstraAttraktioner){
284
285     int i, j, antalIkkeValgteAttraktioner = ANTAL_ATTRAKTIONER - *antalValgteAttraktioner;
286
287     for (i = 0; i < *antalValgteAttraktioner; ++i)
288     {
289         for (j = 0; j < antalIkkeValgteAttraktioner; ++j)
290         {
291             if(atrakktionErTilfoejet(ekstraAttraktioner, *antalEkstraAttraktioner, ikkeValgteAttraktioner[j])){
292             }else if(findDist(ruteAttraktioner[i], ikkeValgteAttraktioner[j], kanter) < maxDist ||
293                 findDist(ruteAttraktioner[i+1], ikkeValgteAttraktioner[j], kanter) < maxDist){
294                 ekstraAttraktioner[*antalEkstraAttraktioner] = ikkeValgteAttraktioner[j];
295                 *antalEkstraAttraktioner += 1;
296             }else{
297                 findEkstraAttraktionerFirkant(ruteAttraktioner[i], ruteAttraktioner[i+1],
298                     kanter, ikkeValgteAttraktioner[j], maxDist, ekstraAttraktioner,
299                     antalEkstraAttraktioner);
300             }
301         }
302     }
303 }

```

```

303 int attraktionErTilfoejet(attraktion *ekstraAttraktioner, int antalEkstraAttraktioner, attraktion
    attraktionAtTilfoeje){
305     int i;
    for (i = 0; i < antalEkstraAttraktioner; ++i)
307     {
        if(strcmp(ekstraAttraktioner[i].navn, attraktionAtTilfoeje.navn) == 0){
309             return 1;
        }
311     }
    return 0;
313 }

315 void findEkstraAttraktionerFirkant(attraktion startAttraktion, attraktion slutAttraktion, kant *kanter,
    attraktion attraktionAtTilfoeje, double maxDist,
        attraktion *ekstraAttraktioner, int *antalEkstraAttraktioner){
317     double vektorLaengde;
    vektor ruteVektor, ruteVinkelretVektor, ruteEnhedsVinkelretVektor, mainHjoerne, side1Vektor,
        side2Vektor, punktVektor;
319
    vektorLaengde = findDist(startAttraktion, slutAttraktion, kanter);
321     ruteVektor.x = slutAttraktion.kmFraGreenwich - startAttraktion.kmFraGreenwich;
    ruteVektor.y = slutAttraktion.kmFraAekvator - startAttraktion.kmFraAekvator;
323     ruteVinkelretVektor.x = -ruteVektor.y;
    ruteVinkelretVektor.y = ruteVektor.x;
325     ruteEnhedsVinkelretVektor.x = ruteVinkelretVektor.x / vektorLaengde;
    ruteEnhedsVinkelretVektor.y = ruteVinkelretVektor.y / vektorLaengde;
327     mainHjoerne.x = startAttraktion.kmFraGreenwich + ruteEnhedsVinkelretVektor.x * maxDist;
    mainHjoerne.y = startAttraktion.kmFraAekvator + ruteEnhedsVinkelretVektor.y * maxDist;
329     side1Vektor.x = ruteVektor.x;
    side1Vektor.y = ruteVektor.y;
331     side2Vektor.x = -2 * ruteEnhedsVinkelretVektor.x * maxDist;
    side2Vektor.y = -2 * ruteEnhedsVinkelretVektor.y * maxDist;
333
    punktVektor.x = attraktionAtTilfoeje.kmFraGreenwich - mainHjoerne.x;
    punktVektor.y = attraktionAtTilfoeje.kmFraAekvator - mainHjoerne.y;
335     if(0 < prikProdukt(punktVektor, side1Vektor) && prikProdukt(punktVektor, side1Vektor) <
        prikProdukt(side1Vektor, side1Vektor) &&
337         0 < prikProdukt(punktVektor, side2Vektor) && prikProdukt(punktVektor, side2Vektor) <
            prikProdukt(side2Vektor, side2Vektor)){
        ekstraAttraktioner[*antalEkstraAttraktioner] = attraktionAtTilfoeje;
339         *antalEkstraAttraktioner += 1;
    }
341 }

343 double prikProdukt(vektor vektor1, vektor vektor2){
    return (vektor1.x * vektor2.x) + (vektor1.y * vektor2.y);
345 }

347 void valgAfEkstraAttraktioner(attraktion *valgteAttraktioner, int *antalValgteAttraktioner, attraktion
    *ekstraAttraktioner, int antalEkstraAttraktioner){
    int i, j = *antalValgteAttraktioner, k, y = 0;
349     int opretNy = 1;

351     if (antalEkstraAttraktioner == 1)
    {
353         printf("\nRuten er blevet fundet. Attraktioner nederfor, der ligger tæt på din nuværende rute,
            kunne give dig en mere interessant tur.\n");
        printf("Tast 1 og tryk enter for at tage attraktioner med. Tast 0, efterfulgt af enter, for at gå
            videre..\n");
355     }
    else if(antalEkstraAttraktioner > 1)
357     {
        printf("\nRuten er blevet fundet. Attraktionerne nederfor, der ligger tæt på din nuværende rute,
            kunne give dig en mere interessant tur.\n");
359         printf("Vælg et tal (svarende til en attraktion) af gangen, efterfulgt af enter efter hver indtastet
            tal.\n");
    }

```

```
printf("Tast 0, efterfulgt af enter, for at gaa videre. Indtast ikke samme tal 2 gange.\n");
361 }
for (i = 0; i < antalEkstraAttraktioner; ++i)
363 {
    printf("%d: %s\n", i+1, ekstraAttraktioner[i].navn);
365 }

367 do{
    opretNy = 1;
369     if (scanf("%d", &k) == 1){
        for (y = 0; y < j; y++)
371         {
            if (strcmp(valgteAttraktioner[y].navn, ekstraAttraktioner[k-1].navn) == 0)
373             {
                printf("Du har allerede indtastet denne attraktion. Proev igen.\n");
375                 opretNy = 0;
            }
        }
        if (k == 0)
377             break;
        else if (k > antalEkstraAttraktioner || k < 0)
379             printf("Tallet svarer ikke til en attraktion\n");
        else if (opretNy && (k <= antalEkstraAttraktioner))
381         {
            valgteAttraktioner[j] = ekstraAttraktioner[k-1];
383             printf("Tilfoejet attraktion: %s\n", ekstraAttraktioner[k-1].navn);
            j++;
385         }
        }
387     }
389     else
    {
        printf("Fejlindtastning - proev igen\n");
        char e[MAX_STRING];
391         scanf("%s", e);
        k = 1;
393     }
395 } while(j < antalEkstraAttraktioner + *antalValgteAttraktioner && k != 0);
397 *antalValgteAttraktioner = j;
}

399 void aendre_startsted(attraktion *ruten, attraktion nytStartSted, int antalAttraktioner, attraktion
    *outputRute)
401 {
    int i = 0, startStedIndex = 0;
403
    for (i = 0; i < antalAttraktioner; i++)
405     {
        if (strcmp(ruten[i].navn, nytStartSted.navn) == 0)
407             startStedIndex = i;
    }
409
    for (i = 0; i < antalAttraktioner; i++)
411     {
        if (startStedIndex == antalAttraktioner-1)
413         {
            startStedIndex = 1;
            outputRute[i] = ruten[0];
415         }
        else
417         {
            outputRute[i] = ruten[startStedIndex];
            startStedIndex++;
419         }
    }
421 }
423 }
425 }
```

```
void outputTilFil(attraktion *ruteAttraktioner, int antalValgteAttraktioner){
427 FILE *filSkabelon, *filOutput;

429 kopierFil(filSkabelon, filOutput);

431 int i;
433 char buf[MAX_STRING];

435 filOutput = fopen("output.kml", "r+");

437 if(filOutput != NULL){
439     while(fscanf(filOutput, " %s", buf) == 1){
441         if(strcmp(buf, "<coordinates>") == 0){
443             fseek(filOutput, 0, SEEK_CUR);
445             fprintf(filOutput, "\n");
447             for (i = 0; i < antalValgteAttraktioner+1; ++i)
449             {
451                 fprintf(filOutput, "%f,%f \n", ruteAttraktioner[i].kmFraGreenwich/KM_PR_LNDGRAD,
453                     ruteAttraktioner[i].kmFraAekvator/KM_PR_BRDGRAD);
455             }
457             fprintf(filOutput, "</coordinates>\n");
459             fprintf(filOutput, "</LineString>\n");
461             fprintf(filOutput, "</Placemark>\n");
463             fprintf(filOutput, "</Document>\n");
465             fprintf(filOutput, "</kml>\n");
467             fseek(filOutput, 0, SEEK_CUR);
469         }
471     }
473 }
475 }else{
477     printf("kunne ikke aabne fil\n"); exit(1);
479 }
481 }

483 void kopierFil(FILE *filSkabelon, FILE *filOutput){
485     char ch;

487     filSkabelon = fopen("outputSkabelon.kml", "r");
489     filOutput = fopen("output.kml", "w");

491     if(filSkabelon == NULL || filOutput == NULL){
493         printf("kunne ikke aabne fil\n"); exit(1);
495     }

497     fclose(filSkabelon);
499     fclose(filOutput);

501     while( ( ch = fgetc(filSkabelon) ) != EOF ){
503         fputc(ch, filOutput);
505     }
507 }
```

Grundlæggende programmering: Søgemaskiner.

Beskrivelse:

På hvilke IT-C hjemmesider finder man information om f.eks. "fodbold", eller oplysning om "eksamensdatoer". Til at svare på et sådan spørgsmål benyttes en søgemaskine. En sådan søgemaskine består dels af teknologi der muliggøre at flere bruger systemet samtidigt (mange bruger f.eks. yahoo samtidigt), og teknologi der muliggøre hurtige søgninger. Disse teknologier kan man lære om i dybden på hhv. webprogrammerings- og algoritmekurser på IT-C. Formålet med dette projekt er at konstruere en simpel søgemaskine. I projektet begrænser vi os til forudsætningerne fra grundlæggende programmering. Hensigten er at projektet skal munde ud i et simpelt program som kan:

1. Indlæse en fil med en masse ord, som "abemor", "katte" og "husvogn".
2. En simpel brugergrænseflade hvorfra der kan spørges om et givet ord var i filen.

For at kunne besvare et spørgsmål hurtigere end at undersøge samtlige ord fra filen, starter projekter med en enkelt forklaring i hvorledes man kan konstruere en simpel hurtig søgerutine. Forslag til varianter af opgaven er velkommen. Formålet med projektet er at de studerende skal opnå erfaring med at konstruere et større program ved at sammensætte og anvende de ting der er blevet undervist i på kurset grundlæggende programmering. Denne erfaring er en stor støtte for en bred vifte af efterfølgende kurser på IT-C: F.eks. hvis man tager et kursus som omhandler estimering af hvor lang tid det tager at udvikle større programmer, eller et teknisk kursus der bygger vider på grundlæggende programmering.

Målgruppe/forudsætninger:

Grundlæggende programmering.

Beskrivelse:

Meget af nutidens viden er spredt rundt over det hele. En søgemaskine gør det muligt, hurtigt at finde det man skal bruge, hvorved der altid skal bruges en specifik søgemaskine. Søgemaskiner skal bruges i mange af nutidens databaser, da det tager for lang tid at søge bit efter bit. Er det muligt at udvikle en søgemaskine der kan lave en hurtig søgning, og stadig finde de korrekte tegn i flere dokumenter af gangen?