

# OSM

## Første opgave

Davy Eskildsen og Mads Lund

February 27, 2012

### 1 Linked list

Argumentet `start` til `append` og `prepend` er af typen `Listnode**`, da en liste er defineret som en pejer til det første element i listen, `Listnode*`. Da `append` og `prepend` kan ændre på det første element i en given liste skal listen derfor defineres demed en ny pejer. Vi kan således ændre den given listepejer ved at dereferere den og sætte den til den nye pejer.

#### (b) Fejl i `remv`

I den udleverede implementationen `remv` var der et problem med betingelsen i `while`-lykken,

```
!match(curr->content) && curr!=NULL
```

Denne betingelse skulle give falsk, hvis `match` returnerer sandt for indeholdet af det aktuelle element, eller hvis elementet er en null-pointer. Problemet er, at `match(curr->content)` bliver evalueret før `curr!=NULL`, hvilket giver en null-pointer exception. Det kan løses ved bare at ombytte de to udtryk.

### 2 Braun tree

I filen `testbrauneq.c` er implementeret test af vores implementation af Braun trees. I `main`-funktionen testes eksplicit for funktionerne `remvL`, `remvR`, `addL` og `addR` på både et tomt træ og et træ med elementer. Antallet af elementer som bliver tilføjet angives med første argument til programmet. Desuden testes `lookup`. Til at verificere testene har vi implementeret hjælpefunktionerne `printaslist` og `drawtree` som er implementeret i `drawtree.c`.

## 3 Genaflevering rettelser

### 3.1 Linked list - Hvis malloc returnerer FALSE

I funktionerne `append` og `prepend` allokeres plads til de nye elementer med funktionen `malloc`. Denne funktion returnerer `NULL` hvis det ikke kan lade sig gøre. Da vi ikke lave nogen checks på returværdien, vil der bliver kastet en null-pointer exception når vi efterfølgende prøver at sætte `content`. Vi har valgt at gøre det på denne måde, da `prepend` og `append` i den given header-fil er `void`-funktioner. Alternativt kunne den fx havde returneret en boolsk værdi eller en fejlkode.

### 3.2 Braun tree

Vi har rettet funktionen `stact_init` til så den initialiseret en stak rigtigt ved at sørge for at pegeren til listen bliver sat til `NULL`.

Vi har også gennemgået vores `del`-funktion i Braun træet og sørget for at denne for fritgivet hukommelsen rigtigt.