Oppgave 1b)

Worst case tilfeller av metodene fra forrige oppgave.

push_back() har ingen iterasjon, og vil derfor ha en konstant tid på 6 steg.

push_front() har heller ingen iterasjon, og vil også ha en konstant tid på 6 steg.

push_middle() itererer gjennom listen helt til den har kommet halvveis i listen. Dette gjør at den vil bruke n/2 steg.

get() itererer gjennom listen helt til den finner noden med riktig id. Deretter returnerer den verdien til noden. Dette gjør at den vil bruke n steg i verste tilfelle.

Oppgave 1c)

Dersom N er begrenset til et mindre antall vil dette gjøre at optimaliseringen av programmet vil ikke ha like mye å si. For eksempel, la oss si at N er begrenset til 100. Da vil ikke optimaliseringen påvirke kjøretiden stort, og eksempelvis lineær og eksponentielle funksjoner vil begge kunne brukes.

Dersom N er begrenset til et stort antall vil optimaliseringen spille en stor rolle for kjøretiden til programmet.

Oppgave 2)

O-notasjon ved et binærsøk er gitt ved O(log(n)).

Get(i) er fn <= n,

For hvert binærsøk vil det også kalles på get(i), så den totale tiden blir tiden n fra get(i) multiplisert med tiden O(log(n)) fra binærsøk. Dette vil gi oss en funksjon for tid, n*O(log(n)).