

Student: Erik Eliasson 19980323–3676
Utbildning: Civilingenjör Spel- och programvaruutveckling
Kurs: DV1490
den 27 maj 2018

Projektuppgift

Innehållsförteckning

Introduktion	3
Syfte och frågeställningar.....	3
Förutsättningar.....	3
Metod	3
Resultat.....	4
Algoritm och datastrukturer.....	4
Schemaläggning av operationer.....	4
Jämförelse mellan uppgift ett och två.....	5
Diskussion	6
Algoritm och datastrukturer.....	6
Schemaläggning av operationer.....	6
Jämförelse mellan uppgift ett och två.....	6
Slutsats	7
Bilaga	8
Körning med 1a	8
Körning med 1b	10
Körning med 2	12

Introduktion

Denna rapport berör schemaläggning av operationer till olika operationssalar med hjälp av "Approximate Bin Packing" och undersöker hur man på ett effektivt sätt kan fylla operationssalarna med operationer för flera dagar.

Applikationen som har skrivit tillsammans med Vendela Eklund med studentakronym "VEEK17".

Syfte och frågeställningar

Syftet med detta projektet var att undersöka hur man på ett effektivt sätt kan utnyttja så kallade "Approximate Bin Packing" för att fylla operationsrum med operationer.

- Vilken algoritm har använts och vilka datastrukturer?
- Vilka eventuella operationer kunde inte schemaläggas, och hur prioriterade algoritmen mellan olika operationer?
- Hur stor andel av tillgänglig tid var resp. operationssal schemalagd?
- Kan operationssalarna utnyttjas på ett bättre sätt än vad din algoritm föreslår?
- Jämför den beräkningstid som krävdes för att finna en tillräckligt bra lösning i uppgift ett med den beräkningstid som krävdes för att finna en tillräckligt bra lösning i uppgift två. Hur effektiv tror du att din algoritm är om antalet operationssalar och operationer som ska schemaläggas ökar ännu mer?
- Fungerar den algoritm du utvecklat och implementerat för att generera en lösning inom rimlig tid?

Förutsättningar

Förutsättningarna för denna undersökningen är att alla föremålen måste vara lagrade offline och inte online.

Metod

Metoden som användes för att undersöka dessa frågeställningarna har varit litteraturstudier av kurs DV1490s kurslitteratur och varit med på föreläsningar om dessa problem. Det har även gjorts ett testprogram som kör algoritmen för 3 olika rum med olika inmatningar för att fastställa att det är korrekt.

Resultat

Algoritm och datastrukturer

Algoritmen som använts fungerar på ett sånt sätt att den rör sig fram och tillbaka mellan de existerande rummen och försöker sprida ut alla operationer mellan rummen. Så ifall där är en summa av operationer som är lika stor som antal rum så kommer det bli en operation per rum istället för att fylla de tillgängliga rummen och försöka hålla så få rum öppna som möjligt.

Algoritmen tar emot en lista av operationer, en array av rum och en int som berättar antalet rum. Anledningen till att dessa datastrukturer är valda är eftersom operationerna vill plockas ut i ordning som en kö men ifall de inte får plats så ska de inte försvinna direkt. Men algoritmen går att anpassa efter att använda en kö utan större förändringar. Anledningen till att rummen sparas i en array och inte listor är eftersom de kommer röra sig fram och tillbaka mellan rummen väldigt mycket och kommer inte flytta på deras position och då är det mer effektivt att använda en array. Följande kommer en pseudokod av algoritmen.

Medan listan av operationer inte är 0

Försök lägg till operation på första platsen i listan i rummet på nuvarande positionen

Om det ej går försök hitta ett annat rum där operationen passar

Ta bort operation oavsett resultat

Rör din position till nuvarande riktning dvs **om** positivt öka med ett, **om** negativ minska med ett. **Ifall** du är på första eller sista rummet så ändra riktningen till korrekt riktning och **om** den är korrekt så rör dig enligt den korrekta riktningen.

För att få olika resultat i denna version av algoritmen så körs algoritmen först utan någon sortering där användaren väljer vilken ordning och därav vilka operationer som är viktigast men vid andra körningen så sorteras de kortaste objekten först och den sista körningen så sorteras de längsta objekten först.

Schemaläggning av operationer

De operationer som inte kunde schema läggas var ofta de som kom sist i sorteringen eftersom oftast hann rummen bli fulla innan det blev deras tur att schema läggas. Algoritmen prioriterar operationer efter deras ordning i listan. Detta kan visualiseras som att man kan lägga olika prioriteringar på operationer som till exempel hade algoritmen i ett verkligt scenario kört akutoperationer först så att de blir oavsett tid schemalagda eftersom det är viktigare att dessa körs så tidigt som möjligt än att vi fyller på med passande långtids operationer och skjuter upp en akutoperation för att få ett optimalt operationsschema.

Fastän algoritmen inte satsade så mycket på att fylla salarna exakt utan lät ordningen på listan vara den bestämmande faktorn för vilken ordning det skulle fyllas på så blev överbliven tid ganska liten.

Speciellt när de långa operationerna var dom som började schemaläggas och sen fyllde man på med små operationer.

Men självfallet kan operationssalarna användas mer effektivt men för det resultatet som blev av denna algoritmen med en så låg kostnad så är denna algoritmen högst relevant eftersom ifall det skulle försöka göra en mer effektiv lösning så ökar risken för att algoritmen kostar för mycket.

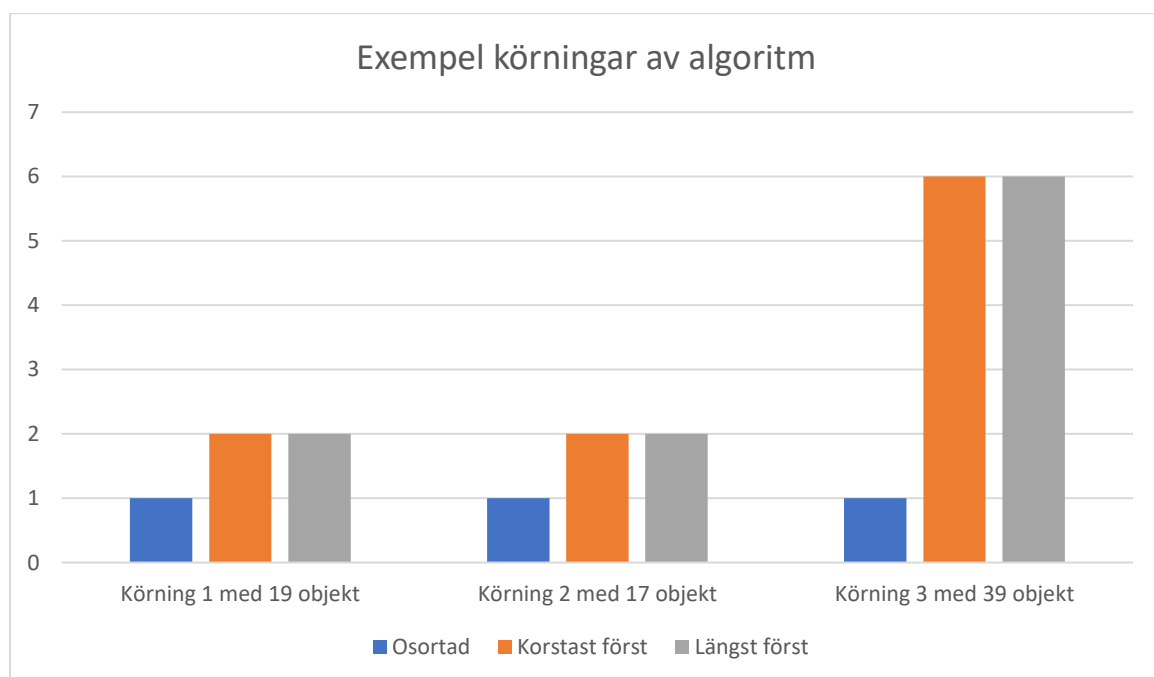
Jämförelse mellan uppgift ett och två

Vid jämförelse av uppgift ett och två så ser man en skillnad i kostnad eftersom antal objekt är större i uppgift två än ett. Kostanden för att det är fler dagar verkar dock inte vara så stor eftersom den osorterade listan tar lika lång tid men vid granskning av kod så märks det att detta borde öka linjärt eftersom den får lite mer arbete att utföra.

Algoritmen i sig är linjär och där borde också synas en lite större kostnad men det som kostar mest är sorteringen eftersom prioritetsskön använder sig av "insertion sort" som vid ett bästa falls scenario är linjär och det nästan alltid är ett bästa falls scenario eftersom elementen i kön sorteras vid inmatning.

En stor anledning till algoritmen är linjär är för att varje gång listan hanteras så är det bara elementet på plats noll som rörs och då blir kostnaden att hämta denna väldigt låg. Det finns inget ställe i koden där listan behöver hämta ett element som inte är det första.

Detta leder till att denna algoritmens starkaste punkt är att den är tidseffektiv samtidigt som den ger ett relevant resultat.



Figur 1. Exempel körning av algoritmen som placerar operationer i operationssalar

Diskussion

Algoritm och datastrukturer

Det man märker av pseudokoden är att algoritmen som används är väldigt baserad på hur listan är sorterad. Eftersom det garanterat att de första operationerna i listan kommer in i en sal men mindre chans att de sista i listan får plats. Därav så är de användaren som bestämmer vilka operationer som är de viktigaste att utföras. Men i de exempel som algoritmen har testats i så ger den där de långa operationerna är de första att schema läggas bäst resultat.

Anledningen till detta är eftersom om det fylls på med de korta operationerna först så när de är ca en timme kvar så har alla operationer under och exakt en timme ändvänts. Då kan vi inte schema lägga mer och det blev en timme oschemalagd operationssal. Men ifall de stora är först så kommer alltid de små operationerna kunna fylla salarna på slutet av dagarna.

Eftersom de korta operationerna fyller salarna när de stora inte får plats längre så finns det väldigt liten risk att de bara blir stora operationer som körs och för varje dag som schemaläggs så kommer några långa och några korta operationer bli avklarade.

Schemaläggning av operationer

Enligt sättet som algoritmen schemalägger operationer så finns det en stor risk att det blir operationer som inte kan schemaläggas trots att det hade funnits möjligheter om man skakade om bland operationerna.

Eftersom den blir så tidseffektiv och ändå ger ett relativt bra resultat speciellt när de största elementen kommer först så är denna typen av algoritm väldigt bra för när man behöver till exempel en grund att jobba ifrån.

En mer avancerad algoritm hade varit rimlig för ett sånt här problem då i ett verkligt scenario så hade man kunnat låta ett program köra i flera timmar för att skapa ett schema. Det ända problemet som uppstår då är akutoperationer och det är där som denna algoritmen blir aktuell. Ifall det blir en schema krock och en ny operation måste in så hade denna algoritmen kunnat snabbt lösa ett rimligt svar för de kommande dagarna som inte ger samma effektivitet men som ger ett snabbt svar med en rimlig lösning.

Jämförelse mellan uppgift ett och två

Vid jämförelse av uppgift ett och två så blev skillnaden inte alls stor och allting fortsatte fungera som tänkt. Varje typ av algoritm tog en märkbar längre tid men det är förväntat när antal objekt ökar med dubbelt så många.

Slutsats

Slutsatsen som kan dras av denna rapporten är att denna algoritmen är väldigt effektiv för att få en låg kostnad för ett väldigt relevant förslag på hur man ska schemalägga operationssalar, speciellt när man schemalägger de stora operationerna först.

Bilaga

Körning med 1a

Run 1:

Room number : 1 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

13 12 7 6 1

Room number : 2 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

17 14 11 8 5 2

Room number : 3 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

19 18 15 10 9 4 3

Run 2:

Room number : 1 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

13 15 2 11 4

Room number : 2 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

7 12 9 14 3

Room number : 3 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

6 10 17 1 18 5

Run 3:

Room number : 1 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

2 15 13 8

Room number : 2 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

11 9 12 7 16

Room number : 3 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

18 14 1 17 10 6

Time for the runs:

Run 1 : 1

Run 2 : 1

Run 3 : 2

Körning med 1b

Run 1:

Room number : 1 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

13 12 7 6 1

Room number : 2 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

11 8 5 2

Room number : 3 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

16 15 10 9 4 3

Run 2:

Room number : 1 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

3 17 10 2 15

Room number : 2 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

14 13 16 11 12

Room number : 3 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

6 4 9 1 8

Run 3:

Room number : 1 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

4 13 5

Room number : 2 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

12 9 17 6

Room number : 3 has operations: This room is open from 8:00 to 19:0 For 1 days

And contains the following operation Ids

day 1 Ids :

2 10 16 3 14

Time for the runs:

Run 1 : 0

Run 2 : 1

Run 3 : 3

Körning med 2

Run 1:

Room number : 1 has operations: This room is open from 8:00 to 19:0 For 2 days

And contains the following operation Ids

day 1 Ids :

19 13 12 7 6 1

day 2 Ids :

30 25 24 18

Room number : 2 has operations: This room is open from 8:00 to 22:0 For 2 days

And contains the following operation Ids

day 1 Ids :

31 20 14 11 8 5 2

day 2 Ids :

38 36 35 32 29 26 23 17

Room number : 3 has operations: This room is open from 8:00 to 17:0 For 2 days

And contains the following operation Ids

day 1 Ids :

21 10 9 4 3

day 2 Ids :

37 34 28 27 22 16 15

Run 2:

Room number : 1 has operations: This room is open from 8:00 to 19:0 For 2 days

And contains the following operation Ids

day 1 Ids :

30 32 34 18 28 31 12 15 21

day 2 Ids :

23 17 13

Room number : 2 has operations: This room is open from 8:00 to 22:0 For 2 days

And contains the following operation Ids

day 1 Ids :

39 24 9 29 2 19 35 8 38 20

day 2 Ids :

6 33 3

Room number : 3 has operations: This room is open from 8:00 to 17:0 For 2 days

And contains the following operation Ids

day 1 Ids :

16 10 26 11 37 1 22 7

day 2 Ids :

14 27 4

Run 3:

Room number : 1 has operations: This room is open from 8:00 to 19:0 For 2 days

And contains the following operation Ids

day 1 Ids :

10 14 5

day 2 Ids :

9 32 24 4 3

Room number : 2 has operations: This room is open from 8:00 to 22:0 For 2 days

And contains the following operation Ids

day 1 Ids :

26 17 23 25

day 2 Ids :

35 31 28 19 11 34 29 16 30 27

Room number : 3 has operations: This room is open from 8:00 to 17:0 For 2 days

And contains the following operation Ids

day 1 Ids :

18 13 6

day 2 Ids :

2 39 33

Time for the runs:

Run 1 : 5

Run 2 : 6

Run 3 : 15