# Teoriuppgifter till labb 5

## Uppgift 1

**Vad skiljer en heuristik från en approximationsalgoritm?**

En **heuristik** ger en lösning som kan leda till en optimal lösning men den garanterar **inte** att lösningen är optimal. En **approximations-algoritm** ger en lösning som är garanterat nära den optimala lösningen.

## Uppgift 2

**Rollbesättningsproblemet i labb 5 är ett minimeringsproblem. Vad är målfunktionen för problemet?**

Enligt labblydelsen så ska vi implementera en valfri heuristik som löser konstruktionsproblemet: “Vilka skådespelare ska ha vilka roller för att lösa rollbesättningsinstansen med så få skådespelare som möjligt?”.

Målfunktionen för problemet är därmed att **hitta minsta antalet skådespelare som behövs för att lösa rollbesättningsproblemet.**

## Uppgift 3

**I labb 5 har så kallade superskådisar införts i rollbesättningsproblemet. Varför behövde superskådisar införas i labb 5?**

I labblydelsen så är det angivet att “endast lösbara instanser kommer att ges som indata, men för att heuristiken i polynomisk tid säkert ska kunna hitta en lösning så är det tillåtet att använda [...]” superskådisar.

Superskådisar har alltså införts i labb 5 för att vi i polynomisk tid ska kunna hitta en lösning till rollbesättningsproblemet.

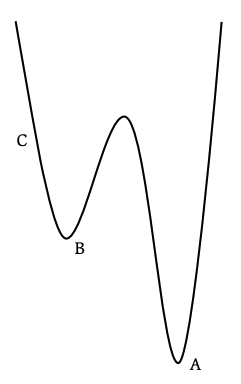
## Uppgift 4

**Om lokalsökning ska användas som heuristik för rollbesättningsproblemet behövs en metod för lokal modifiering av en lösning. Föreslå en lokal modifiering för rollbesättningsproblemet!**

En heuristik som man skulle kunna använda för rollbesättningsproblemet är att ersätta en roll som spelas av skådespelaren p1 och låta någon annan skådespelare p∈{p2, …,pn} spela rollen som spelades av p1. Detta upprepar vi tills alla roller som skådespelaren p1 spelade nu spelas av någon annan skådespelare p∈{p2, …,pn}. Vi kan sedan börja om hela processen med någon annan skådespelare pa∈{p2, …,pn}. Notera: när vi omfördelar rollerna av en viss skådespelare till vissa andra skådespelare så måste omfördelningen av rollerna fortfarande uppfylla kraven för rollbesättningsproblemet, e.g, “varje skådespelare får bara ha en roll i varje scen”.

## Uppgift 5

**Vid lokalsökning gör man lokala modifieringar som inte försämrar målfunktionens värde upprepade gånger. Bevisa att detta tillvägagångssätt (med din lokala modifiering från uppgift 4) inte alltid leder till att den optimala lösningen hittas.**



**Generellt**:

Om man börjar vid startpunkt C och gör lokala modifiering så kommer man till slutpunkt C som är en lokal minimum. Eftersom man vid en lokalsökning inte gör modifieringar som försämrar målfunktionens värde upprepande gånger så leder lösningen inte alltid till att den optimala lösningen hittas (i vårt exempel ovan fastnar vi alltså i punk lokala minimipunkten B.

**Vår lokala modifiering från uppgift 4:**

I vår lokala modifiering så omfördelar vi samtliga roller från en skådespelare till andra skådespelare för att minska antalet skådespelare som behövs. Vår lokala modifiering leder inte alltid till den optimala lösningen eftersom det beror på vilken av skådespelare som tas bort. En skådespelare kan vara avgörande för att hitta den optimala lösningen till indatan. Vi kommer att hitta en lokal minimum men inte alltid det globala minimum.

## Uppgift 6

**Varför fungerar inte ditt bevis i uppgift 5 om Simulated annealing (simulerad härdning) används istället för upprepad lokal förbättring.**

I lokalsökning kan vi inte göra lokala modifiering som försämrar målfunktionens värde. I simulated annealing är det dock tillåtet att göra modifiering som försämrar målfunktionens värde. Tanken är att om vi fastnar i en lokal minima och tillåter ett antal dåliga steg så kan vi komma ur den lokala minimipunkten och förhoppningsvis hitta ett bättre minimipunkt längre fram. Dessutom så använder simulated annealing randomization vilket gör att den inte fastnar i minimipunkter. Simulated annealing är en optimeringsalgoritm som leder till optimala lösningar.

Eftersom simulated annealing tillåter att man tar dåliga steg som försämrar målfunktionens värde samt använder sig utav randomization så kommer den inte att fastna i lokala minima. Vårt bevis i uppgift 5 fungerar inte eftersom simulated annealing är en optimerings-algoritm som leder till optimala lösningar.

## Uppgift 7

**När bör man sluta göra upprepade lokala modifieringar vid lokalsökning? Det vill säga, hur länge ska man hålla på?**

Man kan avsluta lokala modifiering vid lokalsökning när förbättringen från iteration till iteration är mindre än något fördefinierat tal ε. Alternativt så kan man avsluta lokala modifieringar efter en viss tid (time bound) eller ett fixt antal iterationer.

## Uppgift 8

**Vad finns det för fördelar med att införa slump i en heuristik?**

Att införa slump i en heuristik kan förstärka den underliggande heuristiken. En heuristik som har en intern slump mekanism kan undvika värsta falls situationer som en heuristik utan slump mekanismen inte hade undkommit. T.ex, i lokalsökning så skulle vi kunna undvika att fastna i minipunkter genom att införa slump i den lokala modifieringen.