

Anmerkungen zur Tiefensuche und Breitensuche beim CLP

Was sind die wesentlichen Erkenntnisse, die gewonnen werden sollten?

- Ein direktes Lösungsverfahren gibt es für die Problemstellung (möglichst viele Stücke in einen Container / Rucksack gegebener Größe zu packen – möglichst wenige Container für vorgegebene Stücke zu verwenden) nicht, daher ist die Verwendung eines Suchverfahrens zwingend.
- Suchverfahren leisten es, eine im Verlauf der Bearbeitung gefällte Entscheidung, rückgängig zu machen
- Tiefensuche ist ein (ggf. nur potentiell) vollständiges Suchverfahren, bei dem ein eingeschlagener Weg zunächst möglichst weit verfolgt wird und dann mit backtracking so weit zurück gegangen wird, bis noch unverfolgte, alternative Wege wiederum nach demselben Prinzip verfolgt werden können.
- Das Verfahren, bei dem immer zunächst das größte Element eingefügt wird, wenn es passt und keine Entscheidung rückgängig gemacht wird¹, ist zwar sehr effektiv, wenn es Erfolg hat, es hat aber nicht immer Erfolg.
- Ein anderes vollständiges Suchverfahren ist die Breitensuche. Der Unterschied zwischen beiden ist daher nicht, dass nur bei einem von ihnen alle Wege entwickelt werden, sondern in welcher Reihenfolge.
- Tiefensuche mit backtracking ist deswegen so einfach zu programmieren, weil es mit der Datenstruktur stack arbeitet und daher mit Hilfe des Systemstacks leicht zu programmieren ist.
- Breitensuche verlangt daher nicht nur einen neuen Algorithmus, sondern auch eine neue Datenstruktur, weil man nur mit einer queue die zuerst entwickelten Wege weiter entwickeln kann.
- Tiefensuche mit backtracking ist deswegen so erfolgreich bei einfachen Packproblemen, weil die Suchtiefe nicht groß ist und jeweils als Daten nur die Daten des gerade entwickelten Weges gehalten werden müssen.
- Gerade wegen der bekannten Suchtiefe, die in der Regel auch gegangen werden muss, ist Breitensuche bei dieser Problemstellung schlecht bzw. nicht erfolgreich.

Beispiel Stückeliste '(50 50 30 30 30 30) Containerliste '((120) (100))

Vorversuch:

Erste 50 hinein in ersten, zweite 50 auch, erste 30 geht nicht, deswegen in zweiten und zweite und dritte 30 auch, letzte kann nicht eingefüllt werden.

Tiefensuche:

Zunächst wie eben, dann wird (backtracking) die vorletzte 30 entfernt und die letzte versucht, auch nicht besser. Nach etlichen weiteren Schritten wird man soweit zurück gegangen sein, dass die erste 50 nicht in den ersten Container getan wird, mit der zweiten wird es auch nicht besser, schließlich auch diese weggelassen wird, um dann die 30 zu versuchen und nach etlichen weiteren Fehlversuchen zu der passenden Zuordnung zu kommen.

Breitensuche:

Zunächst muss man sich drei Varianten merken, nämlich 50 im ersten Container, im zweiten oder gar nicht eingeladen (je nach Problemstellung kann die letzte Variante weggelassen werden). Dann muss man sich (eigentlich) 9 merken, die sich auf ((120 50 50) (100)) – ((120 50) (100 50)) – ((120) (100 50 50)) – ((120 50) (100)) – ((120) (100 50)) – ((120) (100)) reduzieren ließen. Dann 27 ...

1 ... und damit ist es eigentlich kein Suchverfahren