Första utkast hashtabeller & hashning:

Hashfunktion

separate chaining

linear probing

quadratic probing

double hashing

En hashtabell är ett sätt att lagra data i en typ av array. Hashtbeller används ofta i de fall då det är viktigt att insättning av element och sökning går snabbt. Till skillnad från i t.ex en länkad lista, där varje element måste gås igenom för att hitta den plats där det element man söker efter finns, kan man i en hashtabell snabbt hitta det element man söker efter genom att direkt gå till den plats i arrayen där man vet att det ligger. Vid insättning i en hashtabell, skapas med hjälp av en hashfunktion en kod, som motsvarar ett index i en array där elementet stoppas in. När en sökning efter ett element sedan ska göras, räknas samma kod fram igen, till indexet i arrayen där elementet i bästa fall finns.

Hashkoden skapas med en hashfunktion. Hashfunktionen räknar om delar av elementets data till en kod. En hashfunktionen ska alltid ge samma kod för samma objekt. En mindre effektiv hashkod gör ofta att flera element hamnar på samma plats, medan en mer effektiv hashfunktion gör att elementen sprids ut jämt över hela arrayen, och därmed minimerar risken för att flera element ska få samma kod. Hashning ska ge en kod som är unik i största möjliga mån, så att så få kollisioner som möjligt uppstår. Om två element får samma hashkod, uppstår en kollision, som behöver lösas på något sätt. Ett sätt att hantera kollisioner är linear probing.

Linear probing innebär att då den plats i arrayen som elementen ska in på är upptagen av ett annat element, placeras elementet på den första lediga platsen under det tänkta indexet. Detta kan lätt skapa problem, då den första lediga platsen i värsta falla kan vara i slutet på arrayen. Då elementet sedan ska läsas ur hashtabellen, måste en jämförelse göras med alla element på och under det efterfrågade indexet, tills elementet hittas, vilket kan ta lång tid.

Ett annat sätt att hantera kollisioner är att tillämpa separate chaining. Detta innebär att varje element i arrayen är en pekare till en länkad lista. Om flera element råkar få samma hashkod, länkas de efter varandra i en länkad lista. En sökning efter ett element innebär då att platsen i arrayen först hittas, varpå platsens länkade lista söks igenom efter det önskade elementet.

Quadratic probing är en annan metod för att hantera kollisioner. Det innebär att då det blir en krock, räknas en ny plats ut för elementet genom att ta det ursprungliga indexet 2, och sedan ta modulus med arrayens storlek (för att få en plats som inte hamnar utanför arrayen). Quadratic probing förutsätter att arrayens storlek är ett primtal. Detta för att de nya indexen vid kollision med större sannolikhet blir unika. Tabellen bör inte låtas bli mer än halvfull, eftersom det då inte är säkert att det går att hitta en ledig plats åt element som krockar.

Double hashing är ytterligare en metod för att hantera kollisioner. Double hashing innebär att en hashkod tas fram utifrån elementet som ska placeras. Om en kollision uppstår på denna plats, räknas en ny hashkod ut, för det intervall som ska hoppas fram i arrayen. På denna nya plats kan sedan elementet stoppas om, om den är ledig.