

Course: Data Structures (CSE CS203A, 114-1)
Study Note: Hashing

Student ID: 1133346

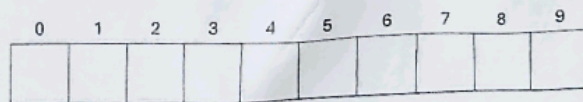
Student Name: 趙婉彤

Definition

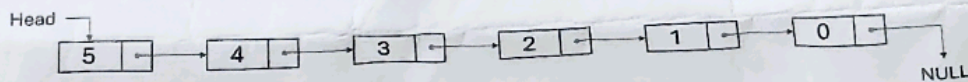
1. Hash Function - A hash function is a function that converts a key (such as a number or string) into an integer index. This index is then used to determine where the data will be stored in a hash table.
2. Hash Table - A hash table is a data structure that stores key-value pairs. It uses a hash function to compute an index in a data structure (called a bucket) where the data will be placed.
3. Collision Handling - A collision occurs when two or more keys are assigned to the same index by the hash function. Collision handling refers to the methods used to store and retrieve these multiple items that share the same index.

Data Structures: Visualization

(1) Array



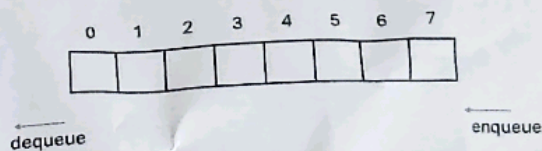
(2) Linked List



(3) Stack

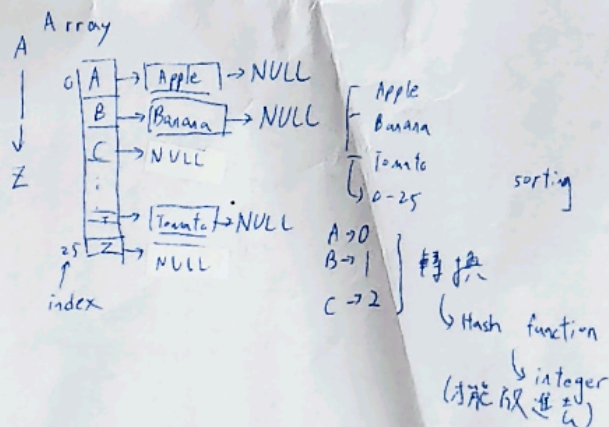
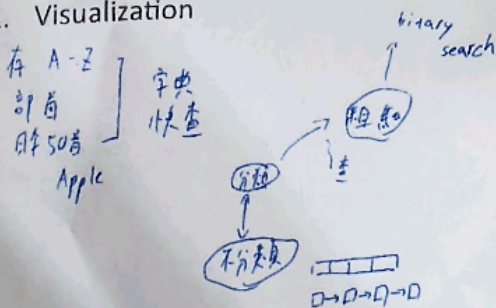


(4) Queue



Note

1. Visualization



2. Abstract Data Type

$\begin{matrix} A=1 \\ B=1 \\ X=25 \end{matrix} \left\{ \begin{matrix} 9 \\ 11 \\ 12 \end{matrix} \right\}$ - Hashing function

sorting \rightarrow order Effective
数字/数字 数字/数字
排序 排序

several pairs with the same key

Hash Table \Rightarrow Array + Linked List

index \leftarrow Hash function
 12, 13, 14
 12 \leftarrow Hash (Apple) $\rightarrow \text{ASCII}('A') - 65 = 0$

$$1 \leftarrow \text{Hash}(\text{Barium})$$

Collision 碰撞

↓ value ⇒ key-value pair

1. 在 array 中發生

key 799 diverse
key 899 888

Hash Function \rightarrow 存取

for example:

Army $\frac{1}{2}$ 23, 33, 43, 53, 63, 73

$$\text{Hash} = \text{input} \bmod 3$$

2 |
3 | → 23, 33, 43, 53, 63, 73

行程 $\frac{1}{2}$ 3

24, 34, 54
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

① the space \Rightarrow 成本 / 投資材料

② capacity → realloc

→ 设计 Hash function → key diverse → collision 减少 / collision 发生
但有 8 个 hex

3. Implementation

Dictionary : characteristic

(collection of word)

2D Array: 很快但浪費空間

Underlying structure

- Array : used for bucket storage

- Linked list : handles collisions

Hybrid: Array + linked list

Array index = bucket (from hash function)

Real-world Application: DNS Caching

collision = different keys, same hash address

probing - collision - resolution technique
open address

open addressing (entire array implementation) - linear probing / quadratic probing / double hashing

chaining

Diagram illustrating the sequence of operations for the first example:

```

graph LR
    A[1] --> B[2]
    B --> C[3]
    C --> D[4]
    D --> E[5]
    E --> F[6]
    F --> G[7]
    G --> H[8]
    H --> I[9]
    I --> J[10]
    J --> K[11]
    K --> L[12]
    L --> M[13]
    M --> N[14]
    N --> O[15]
    O --> P[16]
    P --> Q[17]
    Q --> R[18]
    R --> S[19]
    S --> T[20]
    T --> U[21]
    U --> V[22]
    V --> W[23]
    W --> X[24]
    X --> Y[25]
    Y --> Z[26]
    Z --> AA[27]
    AA --> AB[28]
    AB --> AC[29]
    AC --> AD[30]
    AD --> AE[31]
    AE --> AF[32]
    AF --> AG[33]
    AG --> AH[34]
    AH --> AI[35]
    AI --> AJ[36]
    AJ --> AK[37]
    AK --> AL[38]
    AL --> AM[39]
    AM --> AN[40]
    AN --> AO[41]
    AO --> AP[42]
    AP --> AQ[43]
    AQ --> AR[44]
    AR --> AS[45]
    AS --> AT[46]
    AT --> AU[47]
    AU --> AV[48]
    AV --> AW[49]
    AW --> AX[50]
    AX --> AY[51]
    AY --> AZ[52]
    AZ --> BA[53]
    BA --> BB[54]
    BB --> BC[55]
    BC --> BD[56]
    BD --> BE[57]
    BE --> BF[58]
    BF --> BG[59]
    BG --> BH[60]
    BH --> BI[61]
    BI --> BJ[62]
    BJ --> BK[63]
    BK --> BL[64]
    BL --> BM[65]
    BM --> BN[66]
    BN --> BO[67]
    BO --> BP[68]
    BP --> BQ[69]
    BQ --> BR[70]
    BR --> BS[71]
    BS --> BT[72]
    BT --> BU[73]
    BU --> BV[74]
    BV --> BW[75]
    BW --> BX[76]
    BX --> BY[77]
    BY --> BZ[78]
    BZ --> CA[79]
    CA --> CB[80]
    CB --> CC[81]
    CC --> CD[82]
    CD --> CE[83]
    CE --> CF[84]
    CF --> CG[85]
    CG --> CH[86]
    CH --> CI[87]
    CI --> CJ[88]
    CJ --> CK[89]
    CK --> CL[90]
    CL --> CM[91]
    CM --> CN[92]
    CN --> CO[93]
    CO --> CP[94]
    CP --> CQ[95]
    CQ --> CR[96]
    CR --> CS[97]
    CS --> CT[98]
    CT --> CU[99]
    CU --> CV[100]
    CV --> CW[101]
    CW --> CX[102]
    CX --> CY[103]
    CY --> CZ[104]
    CZ --> DA[105]
    DA --> DB[106]
    DB --> DC[107]
    DC --> DD[108]
    DD --> DE[109]
    DE --> DF[110]
    DF --> DG[111]
    DG --> DH[112]
    DH --> DI[113]
    DI --> DJ[114]
    DJ --> DK[115]
    DK --> DL[116]
    DL --> DM[117]
    DM --> DN[118]
    DN --> DO[119]
    DO --> DP[120]
    DP --> DQ[121]
    DQ --> DR[122]
    DR --> DS[123]
    DS --> DT[124]
    DT --> DU[125]
    DU --> DV[126]
    DV --> DW[127]
    DW --> DX[128]
    DX --> DY[129]
    DY --> DZ[130]
    DZ --> EA[131]
    EA --> EB[132]
    EB --> EC[133]
    EC --> ED[134]
    ED --> EE[135]
    EE --> EF[136]
    EF --> EG[137]
    EG --> EH[138]
    EH --> EI[139]
    EI --> EJ[140]
    EJ --> EK[141]
    EK --> EL[142]
    EL --> EM[143]
    EM --> EN[144]
    EN --> EO[145]
    EO --> EP[146]
    EP --> EQ[147]
    EQ --> ER[148]
    ER --> ES[149]
    ES --> ET[150]
    ET --> EU[151]
    EU --> EV[152]
    EV --> EW[153]
    EW --> EX[154]
    EX --> EY[155]
    EY --> EZ[156]
    EZ --> FA[157]
    FA --> FB[158]
    FB --> FC[159]
    FC --> FD[160]
    FD --> FE[161]
    FE --> FF[162]
    FF --> FG[163]
    FG --> FH[164]
    FH --> FI[165]
    FI --> FJ[166]
    FJ --> FK[167]
    FK --> FL[168]
    FL --> FM[169]
    FM --> FN[170]
    FN --> FO[171]
    FO --> FP[172]
    FP --> FQ[173]
    FQ --> FR[174]
    FR --> FS[175]
    FS --> FT[176]
    FT --> FU[177]
    FU --> FV[178]
    FV --> FW[179]
    FW --> FX[180]
    FX --> FY[181]
    FY --> FZ[182]
    FZ --> GA[183]
    GA --> GB[184]
    GB --> GC[185]
    GC --> GD[186]
    GD --> GE[187]
    GE --> GF[188]
    GF --> GG[189]
    GG --> GH[190]
    GH --> GI[191]
    GI --> GJ[192]
    GJ --> GK[193]
    GK --> GL[194]
    GL --> GM[195]
    GM --> GN[196]
    GN --> GO[197]
    GO --> GP[198]
    GP --> GQ[199]
    GQ --> GR[200]
    GR --> GS[201]
    GS --> GT[202]
    GT --> GU[203]
    GU --> GV[204]
    GV --> GW[205]
    GW --> GX[206]
    GX --> GY[207]
    GY --> GZ[208]
    GZ --> HA[209]
    HA --> HB[210]
    HB --> HC[211]
    HC --> HD[212]
    HD --> HE[213]
    HE --> HF[214]
    HF --> HG[215]
    HG --> HH[216]
    HH --> HI[217]
    HI --> HJ[218]
    HJ --> HK[219]
    HK --> HL[220]
    HL --> HM[221]
    HM --> HN[222]
    HN --> HO[223]
    HO --> HP[224]
    HP --> HQ[225]
    HQ --> HR[226]
    HR --> HS[227]
    HS --> HT[228]
    HT --> HU[229]
    HU --> HV[230]
    HV --> HW[231]
    HW --> HX[232]
    HX --> HY[233]
    HY --> HZ[234]
    HZ --> IA[235]
    IA --> IB[236]
    IB --> IC[237]
    IC --> ID[238]
    ID --> IE[239]
    IE --> IF[240]
    IF --> IG[241]
    IG --> IH[242]
    IH --> II[243]
    II --> IJ[244]
    IJ --> IK[245]
    IK --> IL[246]
    IL --> IM[247]
    IM --> IN[248]
    IN --> IO[249]
    IO --> IP[250]
    IP --> IQ[251]
    IQ --> IR[252]
    IR --> IS[253]
    IS --> IT[254]
    IT --> IU[255]
    IU --> IV[256]
    IV --> IW[257]
    IW --> IX[258]
    IX --> IY[259]
    IY --> IZ[260]
    IZ --> JA[261]
    JA --> JB[262]
    JB --> JC[263]
    JC --> JD[264]
    JD --> JE[265]
    JE --> JF[266]
    JF --> JG[267]
    JG --> JH[268]
    JH --> JI[269]
    JI --> JJ[270]
    JJ --> JK[271]
    JK --> JL[272]
    JL --> JM[273]
    JM --> JN[274]
    JN --> JO[275]
    JO --> JP[276]
    JP --> JQ[277]
    JQ --> JR[278]
    JR --> JS[279]
    JS --> JT[280]
    JT --> JU[281]
    JU --> JV[282]
    JV --> JW[283]
    JW --> JX[284]
    JX --> JY[285]
    JY --> JZ[286]
    JZ --> KA[287]
    KA --> KB[288]
    KB --> KC[289]
    KC --> KD[290]
    KD --> KE[291]
    KE --> KF[292]
    KF --> KG[293]
    KG --> KH[294]
    KH --> KI[295]
    KI --> KJ[296]
    KJ --> KK[297]
    KK --> KL[298]
    KL --> KM[299]
    KM --> KN[300]
    KN --> KO[301]
    KO --> KP[302]
    KP --> KQ[303]
    KQ --> KR[304]
    KR --> KS[305]
    KS --> KT[306]
    KT --> KU[307]
    KU --> KV[308]
    KV --> KW[309]
    KW --> KX[310]
    KX --> KY[311]
    KY --> KZ[312]
    KZ --> LA[313]
    LA --> LB[314]
    LB --> LC[315]
    LC --> LD[316]
    LD --> LE[317]
    LE --
```

↑ key mod n
bucket [index]

Large Array

key \rightarrow index

$$\text{key mod } m \rightarrow \text{sub}$$