

32. GRADUL DE OCUPARE A ZONEI DE MEMORIE

32.1 Zone de memorie

Zona de memorie ocupată de structurile de date include:

- câmpul pentru informația utilă în care se definesc câmpuri ce fac obiectul prelucrărilor;
- câmpul pentru definirea de variabile pointer în vederea stabilirii legăturilor între componente.

Dacă se notează:

- x - numărul de elemente din structura de date;
- Liu - lungimea zonei de memorie ocupată de informația utilă dată ca număr de baiți;
- Lgp - lungimea zonei de memorie ocupată de o variabilă pointer;
- $Lgcpl$ - lungimea zonei de memorie ocupată de variabila pointer spre primul element al structurii de date.

Se obține indicatorul Gru – gradul de utilizare a zonei de memorie, ca raport între lungimea totală a zonei de memorie ocupată de informația utilă în structura de date și totalul zonei de memorie prin care se definește structura de date.

Toate relațiile sunt elaborate pentru luarea în calcul a reprezentării pe 32 de biți pentru variabilele de tip întreg.

Se consideră codul sursă pentru structura care definește informația utilă inclusă în definirea elementelor din lista simplă, lista dublă, arbori, graf, pentru care se calculează gradul de utilizare a zonei de memorie:

```
struct info{
    int cod;
    char nume[30];
    char adresa[40];
    int varsta;
    float salariu;
};
```

În această structură variabila cod este utilizată pentru a identifica fiecare element din colectivitatea căreia i se asociază o structură dinamică. Câmpul cod joacă rol de cheie unică în procedurile de căutare a elementelor și de definire a nodurilor în structura de tip graf.

32.2 Zone de memorie asociate listelor

În cazul *listei simple* cu x elemente, gradul de utilizare este dat de relația:

$$Gru = \frac{x * Liu}{x * (Lgp + Liu) + Lgcpl} \quad (32.1)$$

Pentru lista simplă cu 142 de elemente definite prin:

```

struct element{
    info * inf;
    element * urm;
};

```

rezultă:

- x – numărul de elemente din structura de date este 142;
- Liu – lungimea zonei de memorie ocupată de informația utilă este 78 baiți;
- Lgp – lungimea zonei de memorie ocupată de o variabilă pointer este 2 baiți;
- $Lgcpl$ – lungimea zonei de memorie ocupată de variabila pointer spre primul element al structurii de date este de 2 baiți.

și rezultă că gradul de utilizare pentru lista simplu înlănțuită are valoarea:

$$Gru = \frac{142 * 78}{142 * (78 + 2) + 2} = 0.9748 \quad (32.2)$$

În cazul *listei duble* în care apare în plus variabila pointer pentru referirea elementului precedent, numită *prec* gradul de ocupare a zonei de memorie este dat de relația:

$$Gru = \frac{x * Liu}{x * (2 * Lgp + Liu) + 2 * Lgcpl} \quad (32.3)$$

Dacă se consideră lista dublă având elementele definite prin:

```

struct element{
    info * inf;
    element * urm;
    element * prec;
};

```

rezultă:

- x – numărul de elemente din structura de date este 142;
- Liu – lungimea zonei de memorie ocupată de informația utilă este 78 baiți;
- Lgp – lungimea zonei de memorie ocupată de o variabilă pointer este 2 baiți;
- $Lgcpl$ – lungimea zonei de memorie ocupată de variabilele pointer spre primul și spre ultimul element al structurii de date este în total de 4 baiți, considerați pointeri spre capul de lista dublă, respectiv pointer spre sfârșit de lista dublă.

Se obține gradul de utilizare pentru lista dublu înlănțuită care are valoarea:

$$Gru = \frac{142 * 78}{142 * (78 + 4) + 4} = 0.9508 \quad (32.4)$$

32.3 Zone de memorie asociate arborilor

În cazul *arborilor binari* cu x elemente, gradul de utilizare este dat de relația:

$$Gru = \frac{x * Liu}{x * (2 * Lgp + Liu) + Lgcpl} \quad (32.5)$$

Pentru arborele binar ale cărui noduri sunt definite prin secvența C++:

```
struct element{
    info * inf;
    element * fius;
    element * fiud;
};
```

având 142 de noduri gradul de utilizare a memoriei este:

$$Gru = \frac{142 * 78}{142 * (4 + 78) + 2} = 0.9510 \quad (32.6)$$

Dacă *arborele binar* este organizat pe m niveluri, iar pe fiecare din cele K niveluri se află n_i noduri, numărul total de noduri NrN din structura arborescentă, dat de relația:

$$NrN = \sum_{i=1}^K n_i \quad (32.7)$$

Se determină gradul de utilizare a zonei de memorie în arborele binar organizat pe niveluri ca fiind:

$$Gru = \frac{NrN * Liu}{NrN * (Liu + 2 * Lgp) + Lgcpl} \quad (32.8)$$

Pentru un arbore binar, în care nodurile sunt definite prin secvența de program C++:

```
struct nod{
    info * inf;
    element * fiu_stang;
    element * fiu_drept;
};
```

se consideră:

- x - numărul de noduri din arborele binar este 7 noduri;
- Liu - lungimea zonei de memorie ocupată de informația utilă a unui nod este de 78 baiți;
- Lgp - lungimea zonei de memorie ocupată de variabilele pointer asociate nodului descendent stâng și nodului descendent drept

- este în total de 4 baiți;
- L_{gcpl} - lungimea zonei de memorie ocupată de variabila pointer spre nodul rădăcină este de 2 baiți;
- ceea ce conduce la valoarea gradului de ocupare:

$$Gru = \frac{NrN * Liu}{NrN * (Liu + 2 * Lgp) + L_{gcpl}} \quad (32.9)$$

Rezultă că valoarea gradului de utilizare este:

$$Gru = \frac{7 * 78}{7 * (78 + 2 * 2) + 2} = 0.9479 \quad (32.10)$$

Pentru structura de date *arbore B* de grad m cu x noduri în care structura unui nod este reprezentată în figura 32.1:

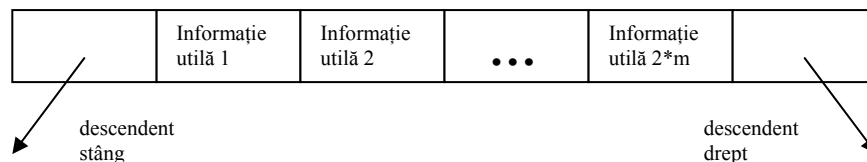


Figura 32.1 Modelul grafic al elementului dintr-un arbore *B*

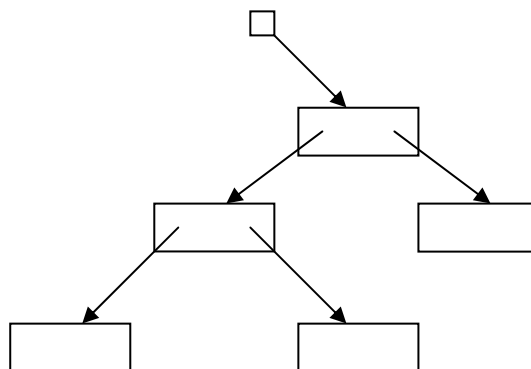


Figura 32.2 Reprezentarea grafică a arborelui *B*

Pentru arborele *B* din figura 32.2 gradul de utilizare al memoriei este:

$$Gru = \frac{x * 2 * m * Liu}{x * (2 * m * Liu + 2 * Lgp) + L_{gcpl}} \quad (32.10)$$

Arborele *B* de ordin m are noduri ce conțin:

- pointer spre descendenții corespunzători informației utile;
- informație utilă;

așa cum sunt definiți în capitolul 13 de arbori *B*. Dacă se consideră definirea nodului din capitolul 13, atunci pentru un arbore *B* a cărui structură este:

```

struct element{
    info * inf[2*m];
    element * desc[2*m+1];
};

```

gradul de utilizare a memoriei se calculează:

$$Gru = \frac{x * 2 * m * Liu}{x * 2 * m * Liu + x * (2 * m + 1) * Lgp + Lgcpl} \quad (32.11)$$

rezultă pentru arborele B de gradul 1:

- x - numărul de noduri din arborele B este 9;
- Liu - lungimea zonei de memorie ocupată de informația utilă a unui nod este de 78 baiți;
- Lgp - lungimea zonei de memorie ocupată de variabilele pointer asociate nodurilor descendente este 2 baiți;
- $Lgcpl$ - lungimea zonei de memorie ocupată de variabila pointer spre nodul rădăcină este de 2 baiți.

ceea ce conduce la valoarea gradului de ocupare:

$$Gru = \frac{9 * 2 * 1 * 78}{9 * 2 * 1 * 78 + 9 * (2 * 1 + 1) * 2 + 2} = 0.9499 \quad (32.12)$$

32.4 Zone de memorie asociate grafurilor

Pentru structura dinamică *graf* definită ca listă de liste, cu o listă principală asociată celor n noduri ale grafului și, respectiv, tot atâtea liste secundare, fiecare având m_i elemente corespunzătoare arcelor incidente spre exterior nodurilor din graf. Se consideră graful din figura 32.3.

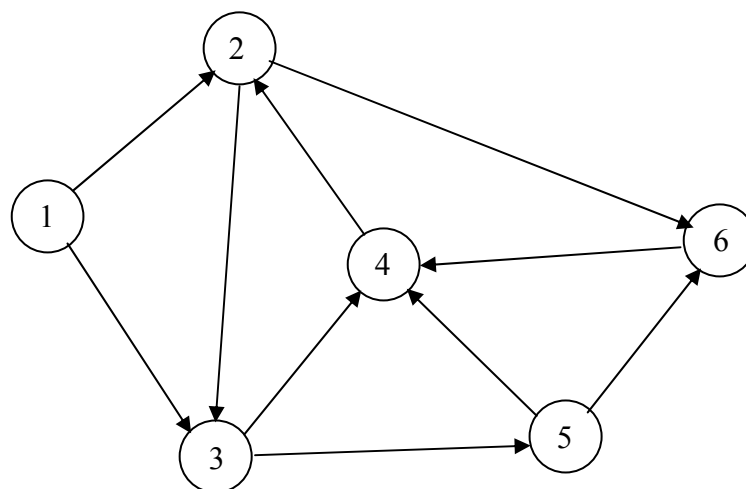


Figura 32.3 Structură de tip graf

Pentru acest graf se construiește lista de liste din figura 32.4.

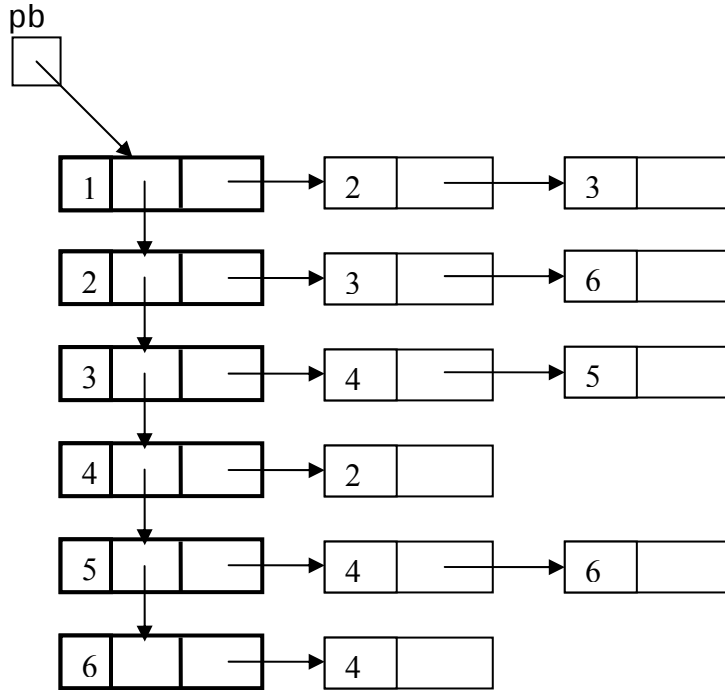


Figura 32.4 Reprezentarea grafului prin liste de liste

Gradul de utilizare a zonei de memorie dintr-o structură de tip graf reprezentat ca listă de liste în care:

- x - numărul de noduri din graf;
- Liu - lungimea zonei de memorie ocupată de informația utilă din fiecare nod și care se memorează în lista de bază; informația utilă conține obligatoriu codul nodului în raport cu care se realizează identificarea arcelor incidente spre exterior;
- $Lgps$ - lungimea zonei de memorie din lista de bază ocupată de variabila pointer de 2 baiți ce reprezintă pointer spre primul element al listei secundare asociată arcelor descendente spre exterior nodului;
- $Lgpb$ - lungimea variabilei pointer de 2 baiți care referă primul element din lista de bază;
- $Lgpp$ - lungimea asociată spre următorul element din lista principală de 2 baiți;
- $Lgspu$ - lungimea pointer-ului spre elementul următor din lista secundară;
- $Lgcd$ - lungimea codului pentru identificarea nodului destinație;
- $Narc$ - numărul total de arce din graf.

$$GruLP = \frac{x * Liu}{x * (Liu + Lgpp + Lgps) + Lgpd} \quad (32.13)$$

$$GruLS = \frac{Narc * Lgcd}{Narc * (Lgcd + Lgspu)} \quad (32.14)$$

$$Gru = \frac{GruLP + GruLS}{2} \quad (32.15)$$

Pentru graful din figura 32.3 ale cărui noduri sunt definite în lista de bază prin secvența C++:

```
struct elemente_Lista_Bază{
    info * inf;
    elementLS * arc;
    elementLP * urm;
};
```

și următoarea structură pentru lista de arce:

```
struct elementLS{
    info * inf;
    elementLS * urm;
};
```

rezultă:

- *x* numărul de noduri din graf este 6;
- *Liu* lungimea zonei de memorie ocupată de informația utilă din fiecare nod este 78;
- *Lgps* lungimea zonei de memorie din lista de bază ocupată de variabila pointer de 2 baiți ce reprezintă pointer spre primul element al listei secundare asociată arcelor descendente spre exterior nodului;
- *Lgpb* lungimea variabilei pointer de 2 baiți care refera primul element din lista de bază;
- *Lgpp* lungimea asociată spre următorul element din lista principală de 2 baiți
- *Lgspu* lungimea pointer-ului spre elementul următor din lista secundară de 2 baiți;
- *Lgcd* lungimea codului pentru identificarea nodului destinație este de 4 baiți;
- *Narc* numărul total de arce din graf este 10

$$GruLP = \frac{6 * 78}{6 * (78 + 2 + 2) + 2} = 0.9473 \quad (32.16)$$

$$GruLS = \frac{10 * 4}{10 * (4 + 2)} = 0.6667 \quad (32.17)$$

$$Gru = \frac{0.9473 + 0.6667}{2} = 0.807 \quad (32.18)$$

În cazul reprezentării *grafurilor* prin vectori de pointeri gradul de utilizare al zonei de memorie se calculează diferit.

Pentru graful definit în figura 32.3, reprezentarea prin vectori de pointeri este:

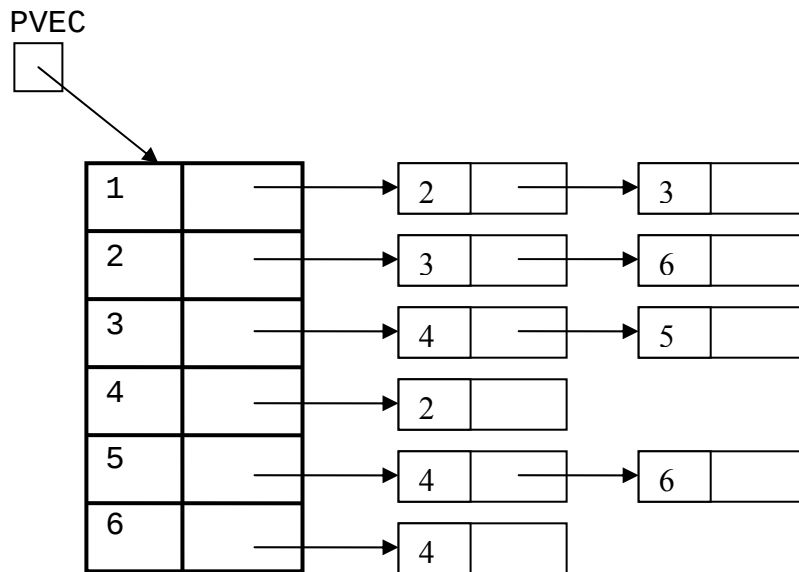


Figura 32.5 Reprezentarea prin vectori de pointeri

rezultă că trebuie luate în considerare următoarele variabile:

- x - numărul de elemente din vectorul de pointeri;
- $Lgve$ - lungimea unui element din vectorul de pointeri care se obține din lungimea informației utile Liu la care se adaugă pointerul spre capul de listă a arcelor incidente spre exterior nodului, adică $Lgve = Liu + Lgp$;
- $Lgpv$ - lungimea variabilei pointer de 2 baiți către vectorul de bază;
- $Lgspu$ - lungimea pointer-ului spre elementul următor din lista secundară de 2 baiți;
- $Lgcd$ - lungimea zonei de memorie în care se stochează codul nodului destinație;
- $Lgpa$ - lungime pointer din lista de arce;
- $Narc$ - numărul de arce din graf.

Gradul de ocupare a zonei de memorie se obține din relația:

$$GruVP = \frac{x * Lgve}{x * Lgve + Lgpv} \quad (32.19)$$

$$GruLS = \frac{Narc * Lgcd}{Narc * (Lgcd + Lgspu)} \quad (32.20)$$

$$Gru = \frac{GruVP + GruLS}{2} \quad (32.21)$$

Pentru graful dat în figura 32.3 cu următoarea structură:

```
struct elementVP{
    info * inf;
    elementLS * urm;
};
```

și următoarea structură pentru lista de arce:

```

struct elementLS{
    info * inf;
    elementLS * urm;
};

```

rezultă ca pentru:

- x - numărul de elemente din vectorul de pointeri egal cu 6;
- $Lgve$ - lungimea unui element din vectorul de pointeri care se obține din lungimea informației utile Liu la care se adaugă pointerul spre capul de lista a arcelor incidente spre exterior nodului, adică $Lgve = Liu + Lgp = 80$ de baiți;
- $Lgpv$ - lungimea variabilei pointer de 2 baiți către vectorul de bază;
- $Lgspe$ - lungimea pointer-ului spre elementul următor din lista secundară de 2 baiți;
- $Lgcod$ - lungimea zonei de memorie în care se stochează codul nodului destinație egală cu 4 baiți;
- $Lgpa$ - lungimea pointerului din lista de arce de 2 baiți;
- $Narc$ - numărul de arce din graf este 10;

gradul de utilizare este:

$$GruVP = \frac{6 * 80}{6 * 80 + 2} = 0.9958 \quad (32.22)$$

$$GruLS = \frac{10 * 4}{10 * (4 + 2)} = 0.6667 \quad (32.23)$$

$$Gru = \frac{0.9958 + 0.6667}{2} = 0.8312 \quad (32.24)$$

Graful din figura 32.2 definit ca *listă de pointeri* are modelul reprezentat în figura 32.6.

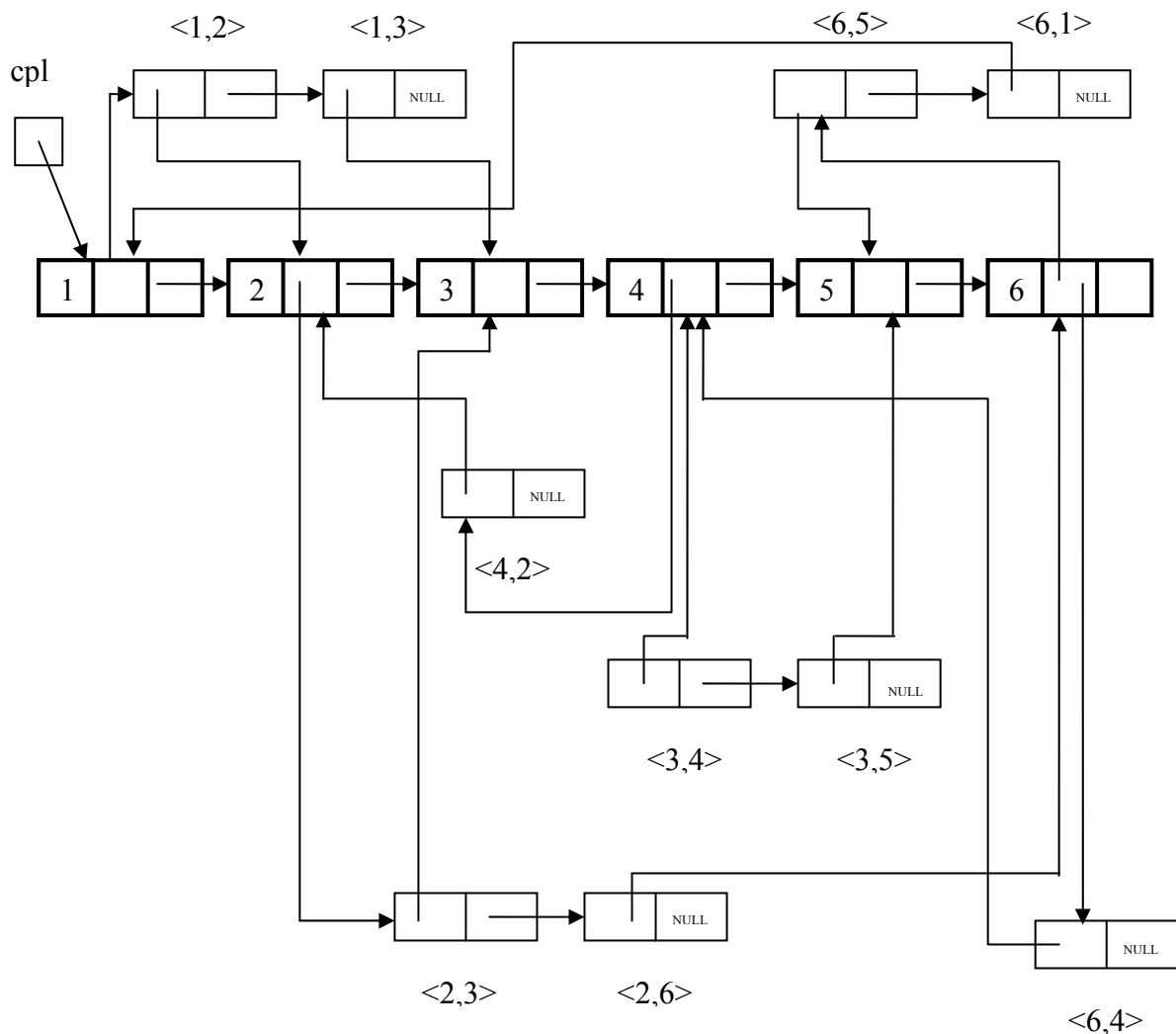


Figura 32.6 Reprezentarea grafului prin listă de pointeri

Se iau în considerare variabilele:

- x - numărul de noduri din graf;
- Liu - lungimea zonei de memorie ocupată de informația utilă din fiecare nod și care se memorează în lista de bază; informația utilă conține obligatoriu codul nodului în raport cu care se realizează identificarea arcelor incidente spre exterior;
- $Lgps$ - lungimea zonei de memorie din lista de bază ocupată de variabila pointer de 2 baiți ce reprezintă pointer spre primul element al listei secundare asociată arcelor descendente spre exterior nodului;
- $Lgpb$ - lungimea variabilei pointer de 2 baiți care refera primul element din lista de bază ;
- $Lgpp$ - lungimea asociată spre următorul element din lista principală de 2 baiți;
- $Lgspu$ - lungimea pointer-ului spre elementul următor din lista secundară;
- $Lgcd$ - lungimea pointerului pentru identificarea nodului destinație;
- $Narc$ - numărul total de arce din graf.

$$GruLP = \frac{x * Liu}{x * (Liu + Lgpp + Lgps) + Lgpb} \quad (32.25)$$

$$GruLS = \frac{Narc * Lgpd}{Narc * (Lgpd + Lgspu)} \quad (32.26)$$

De exemplu, fie următoarea structură:

```
struct elementLP{
    info * inf;
    elementLS * arc;
    elementLP * urm;
};
```

și următoarea structură pentru lista de arce:

```
struct elementLS{
    elementLP * urm;
    elementLS * arc;
};
```

rezultă că pentru:

- *x* - numărul de noduri din graf este 6;
- *Liu* - lungimea zonei de memorie ocupată de informația utilă din fiecare nod și care se memorează în lista de bază; informația utilă conține obligatoriu codul nodului în raport cu care se realizează identificarea arcelor incidente spre exterior este 78;
- *Lgps* - lungimea zonei de memorie din lista de bază ocupată de variabila pointer de 2 baiți ce reprezintă pointer spre primul element al listei secundare asociată arcelor descendente spre exterior nodului;
- *Lgpb* - lungimea variabilei pointer de 2 baiți care referă primul element din lista de bază;
- *Lgpp* - lungimea asociată spre următorul element din lista principală de 2 baiți;
- *Lgspu* - lungimea pointer-ului spre elementul următor din lista secundară este de 2 baiți;
- *Lgcd* - lungimea pointerului pentru identificarea nodului destinație; este de 2 baiți;
- *Narc* - numărul total de arce din graf este 10.

gradul de utilizare este:

$$GruLP = \frac{6 * 78}{6 * (78 + 2 + 2) + 2} = 0.9473 \quad (32.27)$$

$$GruLS = \frac{10 * 2}{10 * (2 + 2)} = 0.5 \quad (32.28)$$

$$Gru = \frac{0.9473 + 0.5}{2} = 0.7236 \quad (32.29)$$

Atunci când se dorește creșterea vitezei de referire a elementelor prin includerea de variabile pointer se va obține scăderea gradului de utilizare a memoriei

În cazul în care se dorește sortarea crescătoare și sortarea descrescătoare a elementelor unei liste simple cu memorarea vechilor poziții ale elementelor, sunt necesari 3 pointeri cap de listă:

- un pointer pentru a referi elementele cu vechea ordine;
- un pointer pentru a referi elementele dipuse în ordine crescătoare;
- un pointer pentru a referi elementele în ordine descrescătoare.

În mod corespunzător, vor fi definiți în fiecare element 3 pointeri pentru a conserva legăturile listei sortate crescător și descrescător, precum și legăturile listei nesorate. Atunci:

$$Gru = \frac{x * Liu}{x * (Liu + 3 * Lgp) + 3 * Lgcpl} \quad (32.30)$$

Pentru a gestiona modificarile din structura se definește o variabilă care are valoarea 0 daca nu s-a produs nici o modificare după realizarea sortării și, respectiv, 1 în caz contrar. Înainte de a se trece la o noua sortare a elementelor din structură se va testa această variabilă. În cazul în care are valoarea 0 sortarea anterioară este considerată operație validă și elementele structurii sunt folosite ca atare. În cazul în care variabila este 1 se va trece la efectuarea unei noi sortări încat structura a suferit modificari în raport cu sortarea precedentă.

32.5 Zone de memorie asociate masivelor

În cazul masivelor definite static cu x elemente, prin una din instrucțiunile:

tip nume[x];

tip nume[m][n];

cu $x = m * n$, gradul de utilizare este:

$$Gru = \frac{x * Liu}{x * Liu} = 1 \quad (32.31)$$

Pentru masivele alocate dinamic, cu x elemente gradul de utilizare este:

$$Gru = \frac{x * Liu}{x * Liu + Lgcpl} \quad (32.32)$$

În cazul unui masiv unidimensional cu 142 de elemente alocat dinamic, avand modelul grafic din figura 32.7.

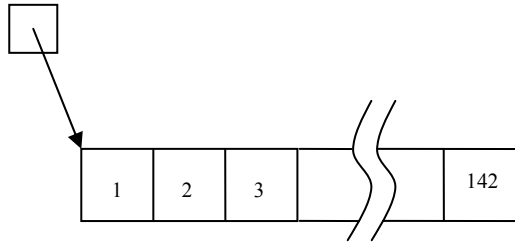


Figura 32.7 Masiv unidimensional alocat dinamic

având definirea:

```
int * vector = new int [142];
```

gradul de ocupare este:

$$Gru = \frac{142 * 4}{142 * 4 + 2} = 0.9965 \quad (32.33)$$

Masivul bidimensional alocat dinamic este caracterizat prin variabilele:

- *Liu* - lungimea informației utile;
- *n* - numărul de linii;
- *m* - numărul de coloane;
- *Lgpm* - lungimea pointerului către primul element al matricii;
- *Lgpvc* - lungimea pointerului către vectorii linie.

având definirea:

```
int ** matrice;

matrice = new int [n];

for(i=0; i<n; i++)
    matrice[i] = new int[m];
```

Gradul de utilizare al memoriei este dat de relația:

$$Gru = \frac{m * n * Liu}{m * n * Liu + n * Lgpvc + Lgpm} \quad (32.34)$$

Pentru un masiv bidimensional alocat dinamic de 142 de elemente, cu modelul grafic dat în figura 32.8.

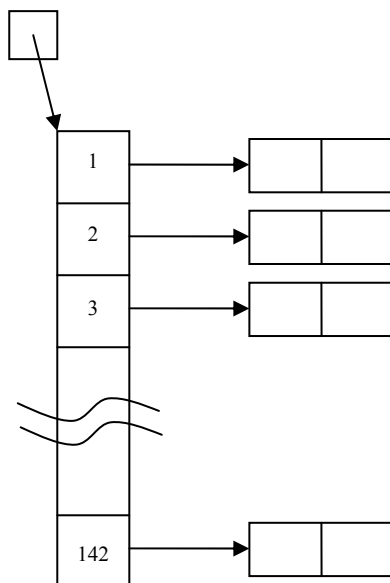


Figura 32.8 Masiv bidimensional alocat dinamic

și cu definirea din secvența:

```
int ** matrice;

matrice = new int [71];

for (i=0; i<71; i++)
    matrice[i] = new int[2];
```

gradul de utilizare este dat de relația:

$$Gru = \frac{2 * 71 * 4}{2 * 71 * 4 + 71 * 2 + 2} = 0.7977 \quad (32.35)$$

Acest indicator, împreună cu indicatorii de performanță ai aplicației informatice privind comportamentul estimat al acestora la utilizator contribuie la completarea imaginii pentru fiecare variantă de program bazată pe o anumită structură de date inclusă în el. Se preferă realizarea unui echilibru între dimensiunea resurselor utilizate, complexitatea expresiilor de referință și viteza de acces la informația solicitată de fiecare beneficiar în parte.