

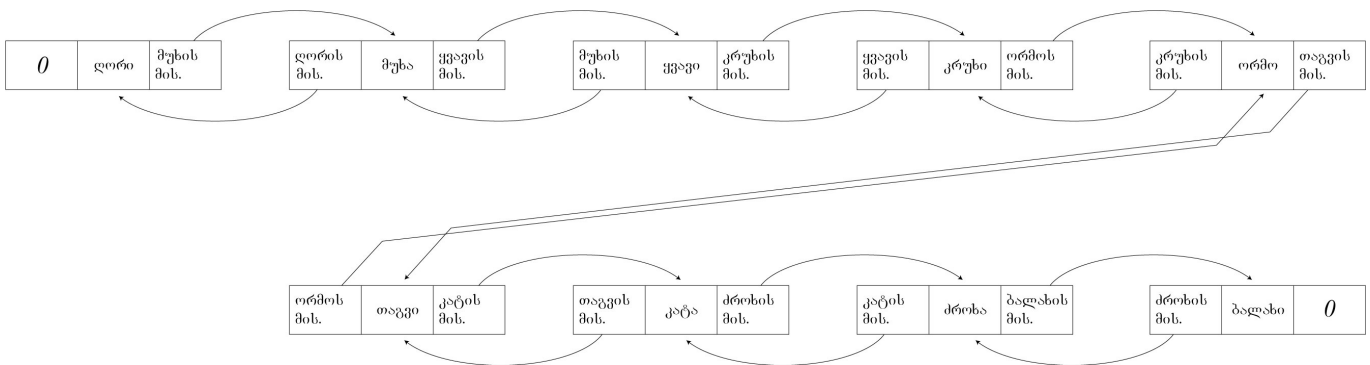
7 ბმული სიები

7.1 რწილი და ჭიანჭველა

ცნობილ ქართულ ზღაპარში „რწილი და ჭიანჭველა“ რწილი ჭიანჭველის გადასარჩენად ღორთან მიდის და თოკისთვის ჯაგარს თხოვს. ღორი მოსთხოვს რკოს და გააგზავნის მუხასთან. მუხა სთხოვს, რომ მოაშოროს ყვავი და გააგზავნის ყვავთან. ყვავი სთხოვს წიწილას და გააგზავნის კრუხთან. კრუხი სთხოვს ფეტვს და გააგზავნის ორმოსთან. ორმო სთხოვს თავის მოშორებას და გააგზავნის თავგთან. თავგი სთხოვს კატის მოშორებას და გააგზავნის კატასთან. კატა სთხოვს რძეს და გააგზავნის ძროხასთან. ძროხა სთხოვს ბალახს და გააგზავნის მინდორში, სადაც რწილი მოკრეფს ბალახს, მიუტანს ძროხას, ის მისცემს რძეს და გააგზავნის უკან კატასთან. კატა რძეს რომ მიიღებს, მოეშევა თავგს და ა.შ.: რწილი განვილიდ ჯაჭვს უკან გაჰყვება, ბოლოს მივა ღორთან, მიუტანს რკოს, მისგან მიიღებს ჯაგარს, დაწინს თოკს და თავის მეგობარს წყლიდან ამოიყვანს.

ამ ზღაპარში ჩვენ ერთი საინტერესო ფაქტი შეგვიძლია დავინახოთ, რომელიც ფართოდ გამოიყენება ინფორმაციკაში:

შექმნილია მონაცემთა ჯაჭვი, რომლის თავშიც დგას ღორი. ღორმა იცის, სად დგას მუხა, ანუ მონაცემთა ჯაჭვში შემდგომი ელემენტის მისამართი. მუხამ იცის ყვავის (ჯაჭვში მისი შემდგომი ელემენტის) მისამართი და ა.შ.: ჯაჭვის ყოველ ელემენტში ჩაწერილია სამი კომპონენტი: ინფორმაცია იმის შესახებ, თუ რა არის ეს კომპონენტი (ჩვენს მაგალითში რა ცხოველი ან მცენარეა), მისი წინა კომპონენტის მისამართი და მისი შემდგომი კომპონენტის მისამართი. გრაფიკულად ეს ნაჩვენებია ნახ. 32. მონაცემთა ასეთ ჯაჭვს ბმული სია ეწოდება. ბმული იმიტომ, რომ ამ ჯაჭვის ყოველი კომპონენტი მის წინა და მის შემდგომ კომპონენტზეა „გადაბმული“: მას თავის წინა და შემდგომი ელემენტის მისამართი აქვს დახსომებული.



ნახ. 32: ზღაპრის „რწილი და ჭიანჭველა“ ბმული სია

აღსანიშნავია, რომ რადგან ღორი ამ სიაში პირველია, მას წინა ელემენტი არ ჰყავს და, შესაბამისად, მისი მისამართიც ვერ ექნება. ამიტომაც წინა ელემენტის მისამართის მაგივრად უწერია 0. ანალოგიური სიტუაციაა ბალახთან: რადგან იგი ბოლო ელემენტია ჯაჭვში, მისი შემდგომი ელემენტის მისამართის ადგილას წერია 0.

აღსანიშნავია, რომ ეს 0 არაა რიცხვის მნიშვნელობის მატარებელი. ეს მხოლოდ იმის მანიშნებელია, რომ ამ ელემენტის შემდგომი (ან წინა) ელემენტი არ არსებობს (ამიტომაც ზოგჯერ წერენ კიდევ , ან ილ იმის აღსანიშნავად, რომ ცვლადი ცარიელია).

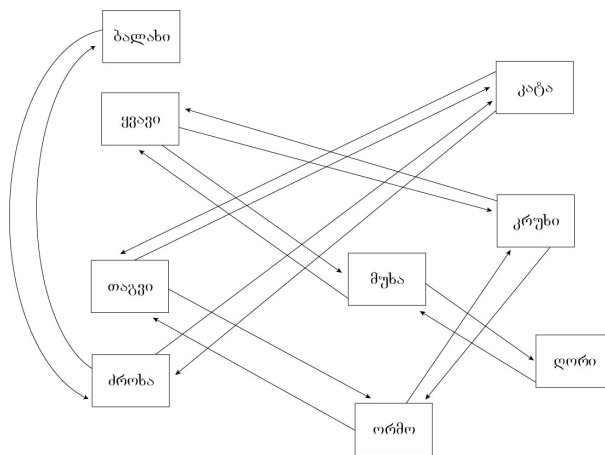
ბმულ სიის ერთ-ერთი უპირატესობაა ის, რომ მასში ელემენტის მოსაძებნად არაა საჭირო ყველა ელემენტის მისამართის ცოდნა, საკმარისია მხოლოდ მისი საწყისი ელემენტის მისამართი ვიცოდეთ: გადავალთ იმ ელემენტთან და თუ ეს არაა ის, რასაც ვეძებთ, გავიგებთ მისი შემდგომი ელემენტის მისამართს. შემდეგ გადავალთ ამ შემდგომ ელემენტზე (თუ ასეთი არსებობს) და ჩავატარებთ იგივე პროცედურას: ვნახავთ, არის თუ არა ეს ის ელემენტი, რომელსაც ვეძებთ. თუ არ არის, ვიგებთ მისი შემდგომი ელემენტის მისამართს (თუ არსებობს) და იგივეს ვიმეორებთ. თუ შემდგომი ელემენტი არ არსებობს, მაშინ საძებნი ელემენტი ამ სიაში არ ყოფილა.

ჩვენს კონკრეტულ მაგალითში, თუ გვაინტერესებს, არის თუ არა კრუხი ამ ბმულ სიაში, ვიწყებთ პირველი ელემენტით, რომლის მისამართი უნდა ვიცოდეთ (ამ შემთხვევაში ამბობენ, რომ საძიებელი ელემენტია „კრუხი“). ვადარებთ მნიშვნელობებს: კატას ვეძებთ, მაგრამ გვხვდება ღორი. ამიტომ ვამოწმებთ, არსებობს თუ არა შემდგომი ელემენტი (ანუ შესაბამის გრაფაში თუ წერია 0). რადგან ამ გრაფაში წერია მისამართი (და არა 0), გადავდივართ ამ მისამართზე. ვადარებთ აქტუალური ჩანაწერის მნიშვნელობას საძიებელი ელემენტის მნიშვნელობას.

აქტუალური ჩანაწერის მნიშვნელობაა „ყვავი“. რადგან ჩვენ ვეძებთ კატას, უნდა გადავიდეთ შემდეგ ელემენტზე, მაგრამ ჯერ შევამოწმოთ, არსებობს თუ არა ასეთი ელემენტი. რადგან შესაბამის გრაფაში არ წერია 0, ამიტომ ასეთი ელემენტი არსებობს და გადავიდეთ მის მისამართზე. ახლა აქტუალური ჩანაწერის მნიშვნელობაა „კატა“. ვადარებთ საძებნ მნიშვნელობას. რადგან ემთხვევა, პროცედურა უნდა დასრულდეს: საძებნი ჩანაწერი ნაპოვნია.

თუ ჩვენს კონკრეტულ მაგალითში გვინდა გავიგოთ, არსებობს თუ არა ჩანაწერი „სპილო“, ისევე ვიწყებთ თავიდან და თუ აქტუალური ჩანაწერის მნიშვნელობა არ ემთხვევა საძებნო ჩანაწერს, გადავიდეთ შემდეგ ელემენტზე, თუ ასეთი არსებობს. როდესაც მივადგებით ჩანაწერს „ბალახი“ და დავასკვნით, რომ იგი არ ემთხვევა საძებნო ჩანაწერის მნიშვნელობას, უნდა გადავიდეთ შემდეგზე, მაგრამ ჯერ შევამოწმოთ, არის თუ არა ეს აქტუალური ჩანაწერი ბოლო მოცემულ ბმულ სიაში. რადგან შემდეგი ჩანაწერის მისამართის შესაბამის გრაფაში წერია 0, შეგვიძლია დავასკვნათ, რომ ეს სიაში ბოლო ელემენტია და პროცედურა დავასრულოთ: ჩანაწერი „სპილო“ ამ სიაში არ გვხვდება.

წინა ნახაზში ბმული სიის მეზობელი ელემენტები ერთი მეორეს მიყოლებით არიან წარმოდგენილი. ეს არაა აუცილებელი: შესაზლოა, რომ ისინი არეულად იყვნენ განლაგებული (ნახ. 33). მთავარია, რომ ყოველმა ელემენტმა მისი წინა და მომდევნო ელემენტების მისამართები იცოდნენ.



ნახ. 33:

ბმული სიის ყოველი ელემენტი სამი კომპონენტისაგან შედგება: ესაა თვითონ ამ კომპონენტის მნიშვნელობა (ანუ გასაღები), მისი ცინა ელემენტის მისამართი და შემდგომი ელემენტის მისამართი. ნახ. 34-ში ზემოთ ნაჩვენებია ეს სამი კომპონენტიანი ელემენტი. თუ იგი განთავსებულია მისამართით x , მისი წინა ელემენტის მისამართი აღინიშნება ფუნქციით $L(x)$ (გრაფიკულად იწერება მარცხენა უჯრაში), ხოლო მისი მომდევნო ელემენტის მისამართი კი აღინიშნება ფუნქციით $R(x)$ (გრაფიკულად იწერება მარჯვენა უჯრაში). თვით მისი მნიშვნელობა აღინიშნება ფუნქციით $Key(x)$ (გრაფიკულად შუა უჯრაში). ამრიგად, $L(კრუხის\ მისამართი) = „ყვავის\ მისამართი“$, $R(კატის\ მისამართი) = „ძროხის\ მისამართი“$, ხოლო $Key(მუხის\ მისამართი) = „მუხა“$.

საგარჯიშო 7.1: რისი ტოლია $L(x)$, $R(x)$ და $Key(x)$, თუ $x = „ძროხის\ მისამართი“$, $x = „თაგვის\ მისამართი“$, $x = „ბალახის\ მისამართი“$, $x = „ორმოს\ მისამართი“$?

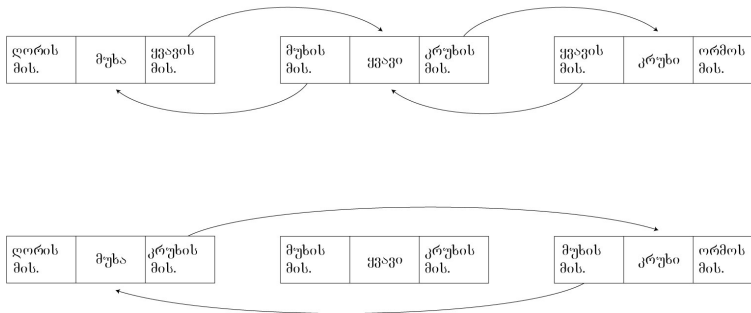
ზოგადად, თუ მოცემულია ბმული სიის რომელიმე ელემენტის მისამართი x , $Key(x)$ ამ ჩანაწერის მნიშვნელობაა, $L(x)$ მისი წინა ჩანაწერის მისამართი, ხოლო $R(x)$ კი - მისი მომდევნო ჩანაწერის მისამართი.

აქედან გამომდინარე, $R(L(x))$ არის x მისამართზე მყოფი ელემენტის წინა ჩანაწერის მარჯვენა გრაფას მნიშვნელობა

საგარჯიშო 7.2: რას ნიშნავს ჩანაწერები $R(R(x))$, $L(L(x))$, $L(R(x))$, $Key(L(x))$ და $Key(R(x))$?

როდესაც ვწერთ $L(x)$, $Key(x)$ ან $R(x)$, იმის და მიხედვით, თუ რისი ტოლია x , ეს ჩანაწერებიც სხვადასხვა იქნება. ამ შემთხვევაში იტყვიან, რომ x აქტუალურ ჩანაწერზე მიუთითებს. როდესაც x ბმული სიის რომელიმე ჩანაწერის მისამართია (ანუ ამ ჩანაწერზე მიუთითებს), მის შემდგომ ელემენტზე „გადასვლა“ (ანუ მის შემდგომ

$L(x)$	$Key(x)$	$R(x)$



ნახ. 34:

ელემენტზე მითითება) შეიძლება ბრძანებით $x = R(x)$. აქ x ცვლადს მიენიჭება აქტუალური ჩანაწერის შემდეგი ელემენტის მისამართი და იგი ამ შემდეგ ელემენტზე მიუთითებს (შემდეგ ელემენტზე „გადავა“).

საეარჯიშო 7.3: რა ბრძანებით უნდა „გადავიდეთ“ აქტუალური ჩანაწერის წინა ელემენტზე?

ცხადია, სანამ გადავალთ წინა ან მომდევნო ელემენტზე, უნდა შევამოწმოთ, არსებობს თუ არა ეს ელემენტი (ანუ აქტუალური ჩანაწერი ბოლო ან პირველი ხომ არაა).

საეარჯიშო 7.4: რა ბრძანებით შეიძლება შემოწმდეს, არის თუ არა x მისამართზე მყოფი ჩანაწერი ბმული სიის ბოლო ან პირველი ელემენტი?

ახლა წარმოვიდგინოთ, რომ გვინდა ზღაპრის გადაკეთება ისე, რომ ამ ჯაჭვიდან ამოვავლოთ ყვავი: მუხა პირდაპირ აგზავნის კრუხთან (ანუ ჩანაწერი „ყვავი“ ამ ბმული სიიდან უნდა ამოვარდეს). ესე იგი, თავიდან მოცემული გვაქვს სიტუაცია, რომელიც ნაჩვენებია ნახ. 34-ში შუაში და გვინდა მივიღოთ სიტუაცია, რომელიც ნაჩვენებია იგივე ნახაზში ქვემოთ. ჩანახერი „ყვავი“, ბმული სიიდან ამოვარდება იმ თვალსაზრისით, რომ ამ სიაში მოძრაობისას ამ ელემენტს ვეღარ წავაწყდებით. ჩანაწერი „ყვავი“ სადღაც კი იარსებებს, მაგრამ იგი ამ სიის ელემენტი აღარ იქნება.

როგორც ხედა ნახაზიდან ჩანს, ყვავის წინა ელემენტი (ამ შემთხვევაში „მუხა“) უნდა მიუთითებდეს „ყვავის“ შემდგომ ელემენტზე, ამ შემთხვევაში ჩანაწერზე „კრუხი“ და პირიქით: ყვავის შემდგომი ელემენტი (ამ შემთხვევაში „კრუხი“) უნდა მიუთითებდეს „ყვავის“ წინა ელემენტზე, ამ შემთხვევაში ჩანაწერზე „მუხა“. აქედან გამომდინარე, თუ გვინდა რაიმე ელემენტის ბმული სიიდან ამოშლა, მისი წინა ელემენტის მარჯვენა გრაფაში უნდა ჩაიწეროს ამ ელემენტის შემდგომი ჩანაწერის მისამართი, ხოლო ამ ელემენტის შემდგომი ელემენტის მარცხენა გრაფაში უნდა ჩაიწეროს ამ ელემენტის წინა ჩანაწერის მისამართი.

ბრძანებებით ეს ასე ჩაიწერება:

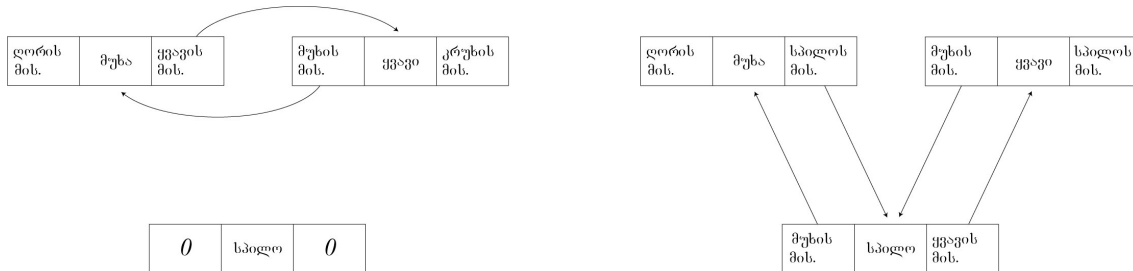
მოც.: x არის იმ ჩანაწერის მისამართი, რომელიც უნდა ამოვშალოთ.

- $R(L(x)) = R(x)$ (აქტუალური ელემენტის წინა ელემენტი უნდა მიუთითებდეს აქტუალური ელემენტის მომდევნო ელემენტზე);
- $L(R(x)) = L(x)$ (აქტუალური ელემენტის მომდევნო ელემენტი უნდა მიუთითებდეს აქტუალური ელემენტის წინა ელემენტზე);
- $x = L(x)$ აქტუალური ელემენტი სიიდან ამოვადებულია. ამიტომ გადავიფაროთ მის წინა ელემენტზე.

საეარჯიშო 7.5: დაუშვათ, x არის ბმული სიის პირველი ელემენტის მისამართი. ბრძანებებით ჩაწერეთ, როგორ შეიძლება ამ ელემენტის ბმული სიიდან წაშლა.

საეარჯიშო 7.6: დაუშვათ, x არის ბმული სიის ბოლო ელემენტის მისამართი. ბრძანებებით ჩაწერეთ, როგორ შეიძლება ამ ელემენტის ბმული სიიდან წაშლა.

ახლა კი წარმოვიდგინოთ, რომ ჩვენი მგალითის ბმულ სიაში ჩანაწერ „მუხასა“ და „ყვავს“ შორის უნდა ჩავსვათ ჩანაწერი „სპილო“ (ნახ. 35 მარცხნივ). საწყის ჯაჭვში ჩანაწერი „მუხა“ შემდგომ ელემენტად მიუთითებს ჩანაწერზე „ყვავი“, ხოლო ჩანაწერი „ყვავი“ წინა ელემენტად მიუთითებს ჩანაწერზე „მუხა“. იმისათვის, რომ ჩავსვათ ჩანაწერი „სპილო“, უნდა შევქმნათ ისეთი ბმულები, როგორიც ნახვენებია ნახ. 35 მარჯვნივ.



ნახ. 35:

ბრძანებებით ეს შემდგენაირად ჩაიწერება:

მოც.: x არის იმ ჩანაწერის მისამართი, რომლის შემდგენაც უნდა ჩაჯდეს ახალი ჩანაწერი (აქტუალური ჩანაწერი); S არის ახალი ჩანაწერის მისამართი.

- $R(S) = R(x)$ (ახალი ელემენტის შემდგომი ელემენტი უნდა იყოს აქტუალური ელემენტის მომდევნო ელემენტი);
- $L(S) = x$ (ახალი ელემენტის წინა ელემენტი უნდა იყოს აქტუალური ელემენტი);
- $L(R(x)) = S$ (აქტუალური ელემენტის მომდევნო ელემენტის წინა ელემენტი უნდა იყოს ახალი ელემენტი);
- $R(x) = S$ (აქტუალური ელემენტის მომდევნო ელემენტი უნდა იყოს ახალი ელემენტი).

საეარჯიშო 7.7: მოცემულია ბმული სია და S მისამართზე განთავსებული ახალი ჩანაწერი. დაწერეთ ბრძანებათა მიმდევრობა, რომელთა საშუალებითაც შეიძლება ახალი ელემენტის სიის პირველ ელემენტად ჩამატება.

საეარჯიშო 7.8: მოცემულია ბმული სია და S მისამართზე განთავსებული ახალი ჩანაწერი. დაწერეთ ბრძანებათა მიმდევრობა, რომელთა საშუალებითაც შეიძლება ახალი ელემენტის სიის ბოლო ელემენტად ჩამატება.

იმისათვის, რომ ვიპოვნოთ ბმული სიის ბოლო ჩანაწერი, უნდა „ვიაროთ მარჯვნივ“ მანამ, სანამ არ შეგვხვდება ბოლო ელემენტი. აქედან გამომდინარე, ეს შეიძლება მოხერხდეს შემდეგი ბრძანებების საშუალებით:

მოცემულია: ბმული სია და მისი ერთ-ერთი ელემენტის მისამართი x .

- while($R(x) \neq 0$)
 $x = R(x)$

საეარჯიშო 7.9: დაამტკიცეთ, რომ ამ ციკლის დამთავრების შემდეგ x ცვლადში სიის ბოლო ელემენტის მისამართი ეწერება.

საეარჯიშო 7.10: დაწერეთ ბრძანებათა მიმდევრობა, რომლითაც ბმული სიის პირველი ელემენტის პოვნა შეიძლება.

საეარჯიშო 7.11: მოცემულია ბმული სია და მისი ერთ-ერთი ელემენტის მისამართი x . აგრეთვე მოცემულია რაღაცა მნიშვნელობა M . დაწერეთ ბრძანებათა მიმღევრობა, რომელთა მეშვეობითაც შეიძლება იმის დადგენა, გვხვდება თუ არა ბმულ სიაში ელემენტის ჩანაწერი, რომლის მნიშვნელობაცაა M (ანუ, სხვა სიტყვებით რომ ვთქვათ, თუ x რაიმე ჩანაწერის მისამართია, $Key(x) = M$).

საეარჯიშო 7.12: მოცემულია n ელემენტიანი ბმული სია. გამოიანგარიშეთ ამ სიაში ელემენტის ჩამატებასა და წაშლისათვის საჭირო ოპერაციათა რაოდენობა.

საეარჯიშო 7.13: მოცემულია ჩვეულებრივი მასივი, რომელიც შედგება n ელემენტისაგან. დაწერეთ ამ მასივში ელემენტის ჩამატების ალგორითმი. გამოიანგარიშეთ მისი ბიჯების რაოდენობა.

საეარჯიშო 7.14: რა უპირატესობა აქვს ბმულ სიას მასივთან შედარებით?