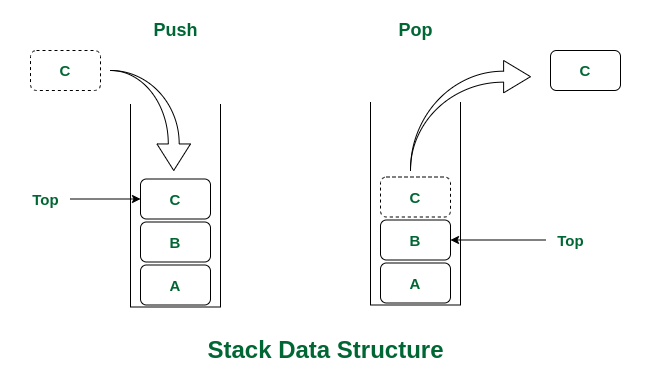
სტეკი

ზოგადად, სტეკი არის მონაცემთა სტრუქტურა, რომელიც მიჰყვება შემდეგ პრინციპს: ბოლოს ჩამატებული მონაცემი, ვარდება პირველი. ამ მონაცემთა სტრუქტურის ბევრი მაგალითი არსებობს რეალურ ცხოვრებაში და გამოიყენება პროგრამირებაში ხშირად. Მაგალითად, ფუნქციების გამოძახებისა და მეხსიერების კონტროლისათვის და სხვა.

პროგრამირებაში და კერძოდ, c++-ში სტეკი, როგორც მონაცემთა სტრუქტურა წარმოადგენს ერთგვარ კონტეინერს, რომელსაც აქვს ორი მთავარი მეთოდი (ფუნქცია): push - ამატებს ახალ ელემენტს სტეკს და pop - შლის ბოლოს ჩამატებულ ელემენტს.

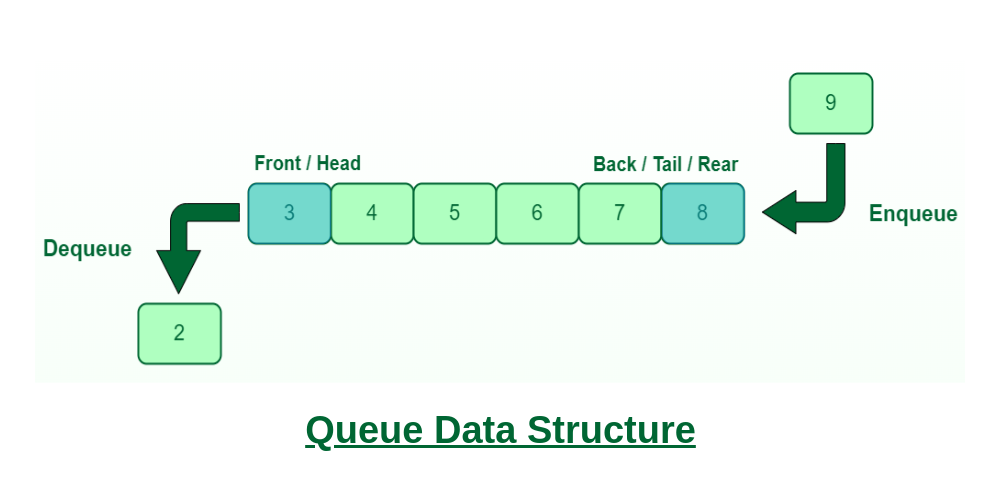
C++ში არსებობს სტანდარტული შაბლონის ბიბლიოთეკა სტეკისათვის “std:stack”, რომელიც ამარტივებს ამ მონაცემთა სტრუქტურის გამოყენებას.



რიგი

რიგი არის ხაზობრივი მონაცემთა სტრუქტურა, რომელიც მიჰყვება შემდეგ პრინციპს: პირველად ჩამატებული მონაცემი ვარდება პირველი. რიგი არის ელემენტების ნაკრები, რომელშიც ელემენტები ემატებიან ერთი ბოლოდან, ხოლო მათი ამოშლა ხდება მეორე ბოლოდან. ამ ფუნქციონალზეა პასუხისმგებელი რიგის ორი ძირითადი მეთოდი enqueue და dequeue. რეალური ცხოვრებიდან რიგის მაგალითად შეგვიძლია ავიღოთ ადამიანების რიგი მაგალითად.

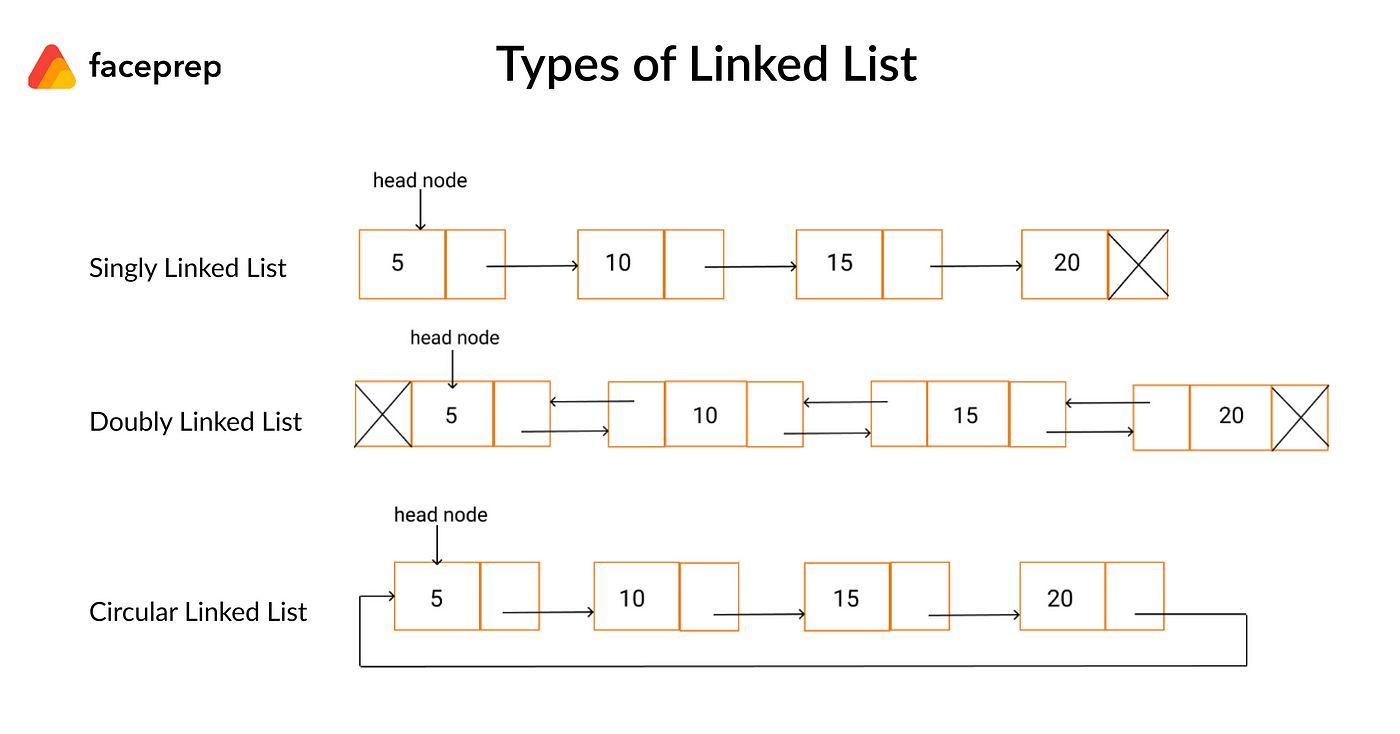
ეს მონაცემთა სტრუქტურა გამოიყენება ხშირად პროგრამირებაში, ალგორითმებში (მაგალითად **Breadth-first search**), რესურსების კონტროლისათვის კომპიუტერებში.



ლისტი

ლისტი არის მონაცემთა სტრუქტურა, რომელიც გამოიყენება პროგრამირებაში მონაცემთა ორგანიზებისა და შესანახად. იგი შედგება ელემენტების თანმიმდევრობისგან, რომელსაც ეწოდება კვანძები, სადაც თითოეული კვანძი შეიცავს ორ ნაწილს: მონაცემებს და მითითებას (ან ბმულს) მიმდევრობის შემდეგ კვანძზე. ბოლო კვანძს, როგორც წესი, აქვს მითითება null-ზე ან სპეციალური მნიშვნელობა სიის დასასრულის მითითებისთვის.

ლისტში მონაცემების დალაგება შეიძლება გარკვეული წესით, შესაბამისად ელემენტის ჩამატებაც მოხდება ამ წესიდან გამომდინარე. ამოშლისას კი საჭიროა კონკრეტული კვანძის იდენტიფიკატორი.



Binary tree

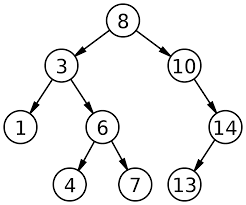
ბინარული ხე არის მონაცემთა იერარქიული სტრუქტურა, რომელშიც თითოეულ კვანძს ჰყავს მაქსიმუმ ორი შვილი, მარცხენა და მარჯვენა. ეს არის ფუნდამენტური კონცეფცია კომპიუტერულ მეცნიერებაში და ფართოდ გამოიყენება სხვადასხვა აპლიკაციებში, მათ შორის მონაცემთა ბაზის სისტემებში, შემდგენელ დიზაინში და სხვა.

ბინარული ხე შედგება კვანძებისგან, რომელიც შეიცავს მონაცემს და მიმთითებლებს მარჯვენა და მარცხენა შვილზე. სულ პირველ კვანძს root ჰქვია.

ბინარული ხე შეიძლება სხვადასხვა პრინციპით იყოს შედგენილი, შესაბამისად არსებობს მისი სხვადასხვა სახე. შესაძლოა დალაგდეს მონაცემები სიდიდის მიხედვით, მშობელი კვანძი დიდი იქნება შვილობილზე და მარცხენა მხარეს მცირე ელემენტები დალაგდება მარჯვენა ელემენტებთან შედარებით.

დაბალანსებული ორობითი ხე არის ხე, რომელშიც თითოეული კვანძის მარცხენა და მარჯვენა ქვეხეების სიღრმე განსხვავდება არაუმეტეს ერთით.

ეს მონაცემთა სტრუქტურა გამოიყენება მონაცემების შესანახად, მათ სწრაფად მოსაძებნად ძებნის ალგორითმის გამოყენებით.



სიმრავლე

კომპიუტერულ მეცნიერებაში სიმრავლე არის მონაცემთა სტრუქტურა, რომელიც წარმოადგენს განსხვავებული ელემენტების დაულაგებელ კოლექციას. მისი მთავარი მახასიათებელია ის, რომ მასში არ შეიძლება იყოს ერთნაირი ელემენტები და ელემენტების შენახვის თანმიმდევრობა არ არის მნიშვნელოვანი.

მასზე შეიძლება ისეთი ოპერაციების განხორციელება, როგორიცაა:

1. ისეთი ელემენტის ჩამატება, რომელიც ჯერ არ არის სიმრავლეში
2. ელემენტის ამოშლა იდენტიფიკატორით
3. შემოწმება, არსებობს თუ არა ელემენტი კონკრეტულ სიმრავლეში
4. ორი სიმრავლის გაერთიანება, თანაკვეთა და სხვაობა.

