

## ლემპელ-ზივის ნაირსახეობა

### შესავალი

ამ დავალებაში, თქვენი მიზანია ლემპელ-ზივის ალგორითმის ერთ-ერთი ნაირსახეობის იმპლემენტაცია. დავალების ფარგლებში უნდა დაწეროთ შემკუმშავიც და გამხსნელიც. კონკრეტული დეტალები ქვემოთ იქნება მოცემული.

მოცემულ ალგორითმს ზუსტად უნდა მიუყვებოდეთ, იმიტომ რომ პროგრამის შემოწმებისას შემკუმშავი და გამხსნელი ცალცალკე შემოწმდება. თუ თქვენს მიერ შეკუმშული ფაილი მოცემული ალგორითმის მიყოლით არ გაიხსნა, ან მოცემული ალგორითმით შეკუმშული ფაილი თქვენი პროგრამით არ გაიხსნა – შესაბამის ნაწილში ქულები აქ ჩაგეთვლება.

ყველა პროგრამას უნდა დაარქვას ის სახელი, რაც შესაბამის საკითხში იქნება მითითებული. დაწერილი პროგრამები (მხოლოდ source-ები!) მოათავსეთ ერთ folder-ში. ამ folder-ს დაარქვით თქვენი freeuni ელ-ფოსტის მისამართი სიმბოლო "@"-მდე. ეს folder-ი შეკუმშეთ zip ფაილად და არქივი ატვირთეთ google classroom-ზე.

### ალგორითმის აღწერა

ფაილის შეკუმშვა მოხდება თითქმის სტანდარტული ლემპელ-ზივის ალგორითმით შემდეგი ცვლილებებით და დამატებებით:

1. შეკუმშული ფაილის დასაწყისში ელიასის გამა კოდით დაიმახსოვრეთ აღსადგენი ფაილის ზომა ბიტებში (რადგან გამა კოდი 1-დან იწყება, ეს მეთოდი ცარიელ ფაილებზე არ იმუშავებს). ლემპელ-ზივის შედეგი ამ გამა კოდს უნდა მოყვებოდეს.
2. თავიდან ლექსიკონში იქნება ორი მიმდევრობა "0" (ინდექსით 0) და "1" (ინდექსით 1).
3. ყოველ ეტაპზე მოცემულ ფაილში (უკვე შეკუმშული ნაწილის გამოკლებით) ვეძებთ ყველაზე გრძელ ფაილის პრეფიქსს რომელიც ლექსიკონში გვხვდება.
4. შეკუმშულ ფაილში ვიმახსოვრებთ არა ლექსიკონში აღმოჩენილი მიმდევრობის ინდექსს და შემდეგ ბიტს, არამედ მხოლოდ ლექსიკონში აღმოჩენილი მიმდევრობის ინდექსს (დამახსოვრებისთვის ჩვეულებრივ ვიყენებთ  $\lceil \log_2(N) \rceil$  ბიტს, სადაც  $N$  ლექსიკონში არსებული მიმდევრობების რაოდენობაა).
5. მოცემული მიმდევრობის ინდექსის შეკუმშულ ფაილში დამახსოვრების შემდეგ, ამ მიმდევრობის მაგივრად ლექსიკონში ორი მიმდევრობა უნდა ჩაიწეროს
  1. ამ მიმდევრობას მარჯვნიდან მიდგმული "0" იმავე ინდექსზე ჩაიწერება რაზეც ეს მიმდევრობა
  2. ამ მიმდევრობას მარჯვნიდან მიდგმული "1" კი ლექსიკონის ბოლოს პირველივე თავისუფალ ინდექსზე (ანუ  $N$ -ზე)
6. როდესაც ფაილის ბოლოს მიხვალთ შეიძლება დაგრჩეთ მიმდევრობა, რომელიც ლექსიკონში არსებული მიმდევრობების მხოლოდ საწყის ნაწილს ემთხვევა. საჭიროების შემთხვევაში დააგრძელეთ ფაილი ზედმეტი ნულების დამატებით. დასაწყისში დამახსოვრებული აღსადგენი ფაილის სიგრძე საშუალებას მოგცემთ ზედმეტი ნულები გადაყართ რადგან აღსადგენი ფაილის სიგრძე გეცოდინებათ.
7. ლემპელ-ზივი აგენერირებს ბიტ სტრინგს, რომელში ბიტების რაოდენობაც შეიძლება არ ემთხვეოდეს 8-ის ჯერადს. ამ მიმდევრობის ფაილში კომპაქტურად დასამახსოვრებლად გამოიყენეთ "დავალება 0"-ის მესამე ნაწილში გაკეთებული მეთოდი.

გამხსნელი პროგრამის ალგორითმი თქვენ თვითონ უნდა შეადგინოთ შემკუმშავის მიხედვით.

### **ჩასაბარებელი ფაილები და ფორმატი**

უნდა დაწეროთ ორი პროგრამა: “LZcompress.xxx” და “LZdecompress.xxx” სადაც xxx თქვენს მიერ არჩეულ პროგრამირების ენაზე დამოკიდებული გაფართოებაა. შემკუმშველი პირველ არგუმენტად უნდა იღებდეს შესაკუმში ფაილის სახელს, კუმშავდეს მას და მეორე არგუმენტად გადმოცემულ ფაილში. გამხსნელი პირველ არგუმენტად უნდა იღებდეს შეკუმშული ფაილის სახელს და აღდგენილ მონაცემებს წერდეს მეორე არგუმენტად გადმოცემულ ფაილში.

### **შეფასება**

თითოეულ პროგრამაში “LZcompress” და “LZdecompress” დაიწერება 4 ქულა (სულ 8 ქულა). იმისთვის რომ ქულა დაიწეროს პროგრამა ზუსტად იგივე შედეგებს უნდა იძლეოდეს რაც სატესტო მონაცემებშია მოცემული. ერთი ბიტის აცდენაც არ არის დასაშვები.

თქვენი პროგრამა არ უნდა მუშაობდეს ძალიან ნელა, და არ უნდა იყენებდეს ძალიან ბევრ მეხსიერებას. კონკრეტულად, პროგრამა შემოწმდება 1 ბირთვიან (საკმაოდ ნელ) პროცესორზე, ნახევარ გიგაბაიტიანი ოპერატიული მეხსიერებით. სატესტო ფაილები იქნება არაუმეტეს 1 მეგაბაიტი ზომის და ამ ზომის ფაილების შეკუმშვაც და გახსნაც, თითო 15 წამზე მეტს არ უნდა მოითხოვდეს. თუ თქვენი პროგრამა შედარებით პატარა სატესტო ფაილებზე სწორად იმუშავებს, მაგრამ დიდ ფაილებზე სამუშაო დროს და/ან მეხსიერებას გადააჭარბებს, დაგეწერებათ შესაბამის ნაწილში მაქსიმალური ქულის 50%.