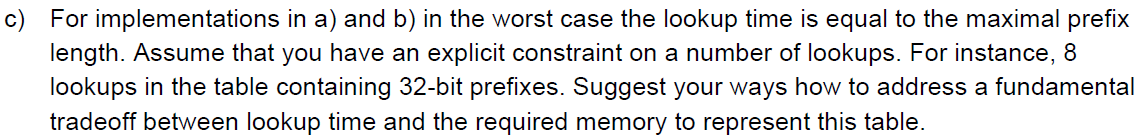
Part 1 - Answers

.



הצעה ראשונה:

route caching in conjunction with linear search to speed up the lookup time. A cache is a fast buffer for storing recently accessed data. The main use of the cache is to speed up subsequent access to the same data if there is a sufficient amount of locality in data access requests.

For lookup the cache stores the recently seen 32-bit destination addresses and the associated next-hop information. When a packet arrives at the router, the destination address is extracted and the route cache is consulted.

Profs: we have 8 lookups only to find, and there is more chance to find it in the Cache of storing the recently accessed data.

Cons: locality exhibited by flows in the backbone has been observed to be very poor. This leads to a much lower cache hit ratio and degenerates to a linear search for every lookup.

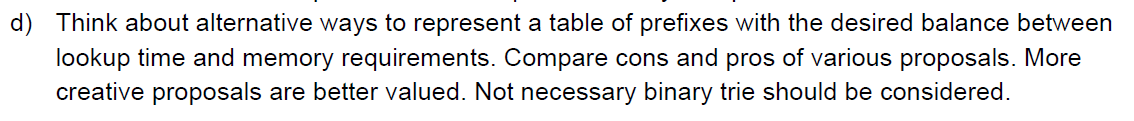
In summary, caching can be useful when used in conjunction with other algorithms, but that precludes the need for fast prefix lookups.

הצעה שנייה עדיפה:

מספר החיפושים במימוש שבו כל צומת בעץ שומרת רק ביט אחד יכול להגיע במקרה הגרוע ל32 קפיצות.

כדיי להקטין את מספר הקפיצות ואת עומק העץ, נשמור בכל נוד 4 ביטים במקום 1, וככה נוכל למצוא את המבוקש בלא יותר מ8 קפיצות.

העומק המקסימלי הוא 4 לעומת 16.



הצעה:

שמירת כל prefix וה action ב hashmap , מתאים לרשתות שאינן מורכבות ובעלות מספר קטן של prefixes\actions, שכן, בעזרת O(1) ישר נוכל למצוא לפרפיקס את הAction המתאים, חשוב לקחת בחשבון שאם יהיו לנו הרבה כאלה אז יהיה שימוש בכמות זיכרון גבוהה.

**Part 1 – input example**

ADD 255.255.128.0/17

11111111.11111111.10000000.00000000

ADD 255.255.0.0/17

11111111.11111111.00000000.00000000

FIND 255.255.5.5

11111111.11111111.00000101.00000101

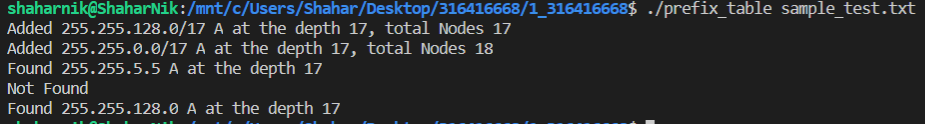
FIND 254.255.255.255

11111110.11111111.11111111.11111111

FIND 255.255.128.0

11111111.11111111.10000000.00000000

**Prefix\_table**



**Prefix\_table\_opt**

