

רקע

עקב הצלחתכם הרבה בתרגילי הבית הקודמים, חברת הצעצועים "DollsRUs" פנתה אליכם וביקשה שתייעצו לה ותסייעו לה בפרסום מודעות של קולקציית הבובות החדשה שלה באינטרנט.

משימה

בתרגיל זה תתנסו בהגשת הצעות לפרסום מודעות מקוונות המופיעות בתוצאות של מנוע חיפוש (sponsored search). עליכם לממש סוכן המקבל נתונים על הגולשים הנכנסים לאתר. בכל כניסה בזאת, הסוכן שלכם מקבל אוסף נתונים רלוונטיים ומגיש הצעה. הצעה זו מתחרה בהצעות של סוכנים נוספים במכרז, ולאחר מכן נבחרים זוכים אשר מודעותיהם מוצגות למשתמשים. אם המודעה שלכם מוצגת, אתם משלמים את ההצעה הבאה אחריכם (לפי GSP עם איכויות q כפי שנלמד בהרצאה). אם גולש לחץ על המודעה שלכם, קיים סיכוי מסוים שהוא יקנה את המוצר שאתם מפרסמים. אם הוא אכן קונה, תרוויחו את סכום הקנייה. הממשק בין הסוכן למנוע החיפוש מתבצע בצורה הפשוטה ביותר ומפורט מטה.

הסבר כללי על הקבצים המצורפים

- id_123456789_987654321.py - קובץ המממש את הסוכנים שלכם. זה הקובץ היחיד בתרגיל הבית שעליכם לשנות, כמפורט בהמשך.
- CONSTANTS.py - קבועים הנדרשים לטובת ההרצה.
- id_dummy_1.py - קובץ המממש סוכן נאיבי.
- id_dummy_2.py - קובץ המממש סוכן נאיבי.
- id_dummy_3.py - קובץ המממש סוכן נאיבי.
- server.py - קובץ הכולל את סימולציית המכרז ואת כלל הפונקציות התומכות הנדרשות.

כאמור, הקובץ היחידי שעליכם לשנות הוא הראשון. זיכרו כי כל שינוי אחר בקבצים יתבצע **לוקלית** אצלכם, ובבדיקת התרגיל נשתמש בקבצי המקור שקיבלתם.

יכולות הסוכן במכרז

כאמור, הקובץ היחיד שעליכם לשנות הינו הקובץ `id_123456789_987654321.py`. הקובץ כולל שתי מחלקות שכל אחת מייצגת סוכן שלכם - סוכן 1 ממומש במחלקה `BiddingAgent1` וסוכן 2 ממומש במחלקה `BiddingAgent2`, כלומר עליכם לממש שני סוכנים. בהמשך, מוסבר מה הייעוד של כל סוכן. על כל סוכן לתמוך ב-3 פונקציות בסיסיות:

- `get_bid` - מקבל את מאפייני המכרז ומחזיר את ההצעה. כל מכרז מבוסס על משתמש שהגיע אל המערכת, ומאפייני המכרז משתנים בהתאם לזהות המשתמש. מאפייני המכרז הינם:
 - מספר הסוכנים המשתתפים במכרז.
 - P - רשימת הסתברויות לקליק של המשתמש הנוכחי, כאשר $P[i]$ היא ההסתברות לקליק על המיקום ה- i .
 - q - ציון איכות המודעה של הסוכן שלכם ביחס למשתמש הנוכחי. הציון הזה הוא ההסתברות המשתמש שהגיע אליכם לאתר יבצע רכישה.
 - v - הערך שלכם במידה והמשתמש לחץ על המודעה (ללא תלות במיקום) ובנוסף החליט גם לקנות את המוצר. שימו לב - אם המודעה שלכם פורסמה במקום ה- i אזי תוחלת ההכנסה שלכם (לפני קיזוז התשלום) היא $P[i]q v$.
- `notify_outcome` - פונקציה אשר נקראת בתום המכרז. הפרמטרים שהיא מקבלת הם:
 - `reward` - הרווח מהמשתמש במידה והוא לחץ על המודעה וביצע רכישה, או 0 אם לא עשה זאת.
 - `payment` - התשלום שלכם שנקבע על פי מכרז GSP.
 - `position` - המיקום בו המודעה שלכם פורסמה. אם לא נבחרתם להציג מודעה באותו מכרז, המיקום יהיה -1.
- `get_id` - פונקציה המחזירה מחרוזת עם תעודות הזהות שלכם.

שימו לב, אין לשנות את שמות הפונקציות או המחלקות הנתונות!

מימוש סוכן אישי

בקובץ `id_123456789_987654321.py` ישנן הערות TODO שהן המשימות שלכם:

1. לשנות את שם הקובץ כך שיתאים לתעודות הזהות שלכם. כך למשל אם אליס ובוב מגישים עבודה ביחד ותעודת הזהות של אליס הינה "111" ושל בוב "222", עליהם להגיש קובץ בשם `"id_111_222.py"`. אם צ'ארלי מגיש את העבודה לבד ותעודת הזהות שלו הינה "333", אז עליו להגיש את הקובץ בשם `"id_333.py"`. שימו לב - הקפידו להשאיר את התחילית "id" בשם הקובץ.
2. מימוש `get_bid` לכל סוכן. המימוש הקיים פשטני, אך שינוי קל שלו יספיק לכם בכדי לנצח את הסף הנמוך.
3. מימוש `notify_outcome` לכל סוכן. אין חובה לשנות את הפונקציה הזאת. אם `get_bid` שמימשתם אינה משתמשת בנתוני מכרזי העבר, גם אין סיבה.

4. מימוש `get_id` לכל סוכן. המחזרות שהפונקציה מחזירה צריכה להיות זהה לשם הקובץ אותו אתם מגישים (כמובן ללא ".py").

שימו לב – החישוב שלכם המתבצע בכל פעם שהפעולה `get_bid` נקראת, צריך להתבצע בסיבוכיות מקום וזמן $O(n)$ כאשר n הוא מספר הסוכנים הקיימים. אין לשנות שום דבר אחר בקובץ, כמו לדוגמא את שם ה-class.

פרוטוקול התחרות

התחרות מבוססת על הפונקציה `server.sequential_game`. אנא קראו את ההסבר הבא עד להבנה מלאה, וכן עברו על הקוד בקובץ המתאים.

הסבר הקוד בקצרה:

קוראים את שמות כל הקבצים בתקיית ההרצה, ובודקים עבור כל קובץ המתחיל ב " <code>_id</code> " האם הסוכן שלו תקני (אם <code>is_agent2=True</code> אז נבחר בסוכן השני בקובץ שלכם. אחרת, נבחר בראשון). אלו הסוכנים לריצה ^[1] .	<code>all_agents = get_agent_list(is_agent2)</code>
לכל סוכן מאותחל משתנה עזר שעוקב אחרי הרווחים, התשלומים, זמן הריצה וכו'.	<code>book_dict = {a: BookKeeping(a) for a in all_agents}</code>
מתוך כלל הסוכנים, נבחרת תת קבוצה שלהם. בחירה אקראית זו מדמה רלוונטיות/אי רלוונטיות של סוכנים מסויימים עבור משתמש נתון. הבחירה נעשית לפי הלוגיקה בפונקציה <code>sample_relevant_agents</code> .	<code>relevant_agents = sample_relevant_agents(all_agents)</code>
כלל המשתנים עבור המכרז הנוכחי מוגרלים ^[2] .	<code>v_list, q_list, prob_click = sample_variables(num_of_agents)</code>
מעבירים את המידע לכל אחד מן הסוכנים הרלוונטים ומבקשים את ההצעה.	<code>bids.append((i, agent.get_bid(num_of_agents, prob_click, q_list[i], v_list[i])))</code>
מריצים מכרז GSP לפי הפונקציה <code>gsp</code> .	<code>gsp_outcome = gsp(bids, q_list, len(prob_click))</code>
עבור כל סוכן שקיבל מיקום, מוגרל מ"מ המציין אם המשתמש הקליק על המודעה או לא.	<code>rewards = sample_rewards(gsp_outcome, q_list, v_list, prob_click)</code>
הסוכן מקבל את האינפורמציה אודות הזכייה.	<code>agent.notify_outcome(reward, payment, position)</code>
סוכנים איטיים מוצאים מהמכרז ולא יבחרו עוד ^[3] .	<code>to_remove = get_timeout_agents(relevant_agents, book_dict)</code>

[1] ודאו כי שלושת הסוכנים הנאיבים (`id_dummy_1` וכו') והקובץ שלכם, הם הקבצים היחידים המתחילים ברישא "`_id`" באותה תיקייה בה קובץ ה-server נמצא.

[2] שימו לב להתפלגויות הללו – ראו הסבר נוסף מטה בחלק "הנחות הסתברותיות".

[3] סוכן יחשב איטי אם ממוצע זמני הריצה שלו גדול מ-`CONSTANTS.TIME_CAP`. השארנו מרווח בטחון די גדול, כך שלא אמורה להיות בעיה עבורכם.

הנחות הסתברותיות

משתני המכרז נדגמים לפי הכללים הבאים:

- קבוצת הסוכנים הרלוונטיים בכל הרצה נדגמים כך שכל סוכן הינו רלוונטי בהסתברות 8% (במידה ויש הרבה סוכנים). בכל מקרה, יהיו לפחות שלושה סוכנים במכרז.
- ההסתברויות לקליק P נדגמות באופן הבא:
 - נדגמים שני מ"מ
 - $d \sim \text{Normal}(0.6, 0.05)$, $T \sim \text{Normal}(0.7, 0.1)$
 - ההסתברות לחיצה על המקום ה- i אשר מסומנת ב- $P[i]$ הינה:

$$P[i] = T \cdot d^{i-1}$$
- האיכות של כל סוכן ביחס למשתמש נתון מורכבת מארבע קטגוריות – גבוה, בינוני, נמוך וטעות. ההסתברות לרכישה הינה: (בעיגול ל-0 ו-1 במקרה של חריגה מהאינטרוול)

$$q \sim \begin{cases} \text{Normal}(0.75, 0.08) & w.p. 0.2 \\ \text{Normal}(0.45, 0.2) & w.p. 0.4 \\ \text{Normal}(0.3, 0.12) & w.p. 0.35 \\ 0.0001 & w.p. 0.005 \end{cases}$$
- הערך של כל סוכן מתפלג לפי $v \sim \text{Normal}(14, 2.5)$, ומעוגל ל-0 במקרה שנדגם ערך שלילי.

מבנה הציון

הציון שלכם מבוסס על שתי משימות:

1. משימה בסיסית המהווה 90% מהציון. במשימה זו הסוכן **הראשון** שלכם יתחרה בשלושת הסוכנים הנאיביים בקבצים שקיבלתם למשך מיליון סיבובים (הקוד שקיבלתם מוכן לבצע סימולציה על משימה זו). במשימה זו אתם נמדדים על פי התועלת הממוצעת אותה השגתם בתום התהליך.
 - נאיבי – השווה 90% מחלק זה - ציון בינארי, עובר או לא. הסוכן שלכם יעבור משימה זו במידה וישיג ציון גבוה יותר משלושת הסוכנים הנאיביים.
 - תחרותי – תקבלו ניקוד נוסף של 10% מחלק זה (פלוס בונוסים) על סמך ההצלחה שלכם מול הסוכנים הנאיביים ביחס לסטודנטים האחרים.

הערה: כדי לנצח את הסוכנים, כדאי להבין איך הם פועלים.
 2. משימה תחרותית המהווה 10% מהציון פלוס בונוסים. במשימה זו הסוכן **השני** שלכם יתחרה בסוכנים של הסטודנטים האחרים למשך מיליון סיבובים. במשימה זו אתם נמדדים על פי התועלת הממוצעת אותה השגתם בתום התהליך.
- כך לדוגמא, אם אליס צלחה את החלק הנאיבי במשימה הבסיסית, היא תקבל ציון של לפחות $81 = 0.9 \cdot 90$.

הוראות נוספות

- ודאו כי שיניתם את שם הקובץ `id_123456789_987654321.py` לפי ההוראות מעלה.
- אין להשתמש בספריות חיצוניות.
- בדאי ואף רצוי לעבור על כלל הקובץ `server`.

הוראות הגשה

יש להגיש שני קבצים בתוך תקיית `zip`:

- קובץ הסוכן שלכם. ודאו כי עקבתם אחר ההוראות המופיעות ב-"יכולות הסוכן במכרז". חריגה תהווה פסילה וציון 0.
- קובץ `pdf` המתאר מילולית את המימוש שלכם עבור הפונקציה `get_bid`. אין להגיש יותר מעמוד אחד. אם המימוש שלכם פשוט מאוד (כמו זה שהקובץ הגיע איתו), גם משפט אחד יכול להספיק.

בהצלחה !