Some notes:

(Regret)

- Is what you could have obtained but didn't - or, to put it more formally, the diff bet the expected cumulative return of an optimal policy and the actual sum of rewards you got.

- Smaller regret means that the policy is better at exploration.
- Regret estimates got NOTHING to do with how quickly a given exploration strategy converges.

(ε-Greedy)

- With constant ε, ε-greedy exploration has linearly growing regret.
- Assuming 't' is the total # actions taken and you set $\varepsilon = 1/t$, an ε-greedy strategy will reach optimal policy in the limit.

(Uncertainty-based Exploration)

- In case of a simple multi-armed bandit, Thompson Sampling has asymptotically smaller regret than
an ε-greedy strategy with $\varepsilon=0.5$
In fact, ε-greedy will end up with linearly increasing regret while Thompson Sampling will be logarithmic.

- In some cases, ε-greedy strategy with $\varepsilon=0.2$ can sometimes have smaller regret than Thompson Sampling by 100-th.
In fact, a random strategy can do that. Because everything is random.
Thompson Strategy is still better asymptotically and in expectation.

(Planning)

- The main diff bet Background planning and Decision-time planning is that the latter plans to select optimal action only for the current state, whereas the former builds a policy that is good in every state.
- Decision-time planning includes alg that can work with both: generative and distributive models.
- Decision-time planning may be much more applicable to the real-world scenarios. In the real-world, a robot may face only subset of all possible states. So it is no point in wasting computational resources to build a good policy in all states, as the Background planning does.

(Heuristic Search)

- Is unreliable in case the model of the world is NOT sufficiently accurate. Also the time required to end up with a good plan crucially depend on the heuristic quality.
- It's NOT specific to RL.
- Is an improvement over full-width search, focusing on the most valuable (according to heurisitc) directions first.
- Is an umbrella term for many alg that make lookahead search based on some heuristic (either hand-crafted or estimated)

(Action Selection based on MC estimates of Q-Fn)

- Acting greedily wrt the estimates of Q-Fn is strikingly similar to VI step.
One step of Value Iteration to be exact.
- We may want to distribute # rollouts unevenly bet diff actions, paying more attention to the most promising ones. That may require exploration techniques.

- Rollout Policy should sacrifice quality of action selection for speed. The reason is simple: the quality can be easily improved by increasing # returns averaged. (WRONG STATEMENT)
By increasing # rollouts, we can improve the quality in a sense of reducing the randomness of the Q-Fn estimate. Recall that Q-Fn is defined wrt some policy and if that policy is bad (or random). its Q-Fn values are low, and improvement over such policy may be very far from optimal.

- Rollout Policy should sacrifice speed for quality of action selection. If each rollout is made by the optimal policy, it is sufficient to make only one rollout per action for the improved policy to be optimal (WRONG STATEMENT)
The env may be stochastic. Thus the one sample estimate of q_ may be built on outlier samples, leading to suboptimal policy that is an improvement upon the optimal rollout policy.

(Selection Phase in MCTS)

- Is the first phase performed by MCTS alg in the very beginning of planning, where we need to descend the alrardy built tree till the leaf node. The root node is not a leaf node.
N.B. in the root state we need to select an action, according to the tree policy.

(Planning in RL)

- Planning computationaly intensive
- Planning alg that can output a valid policy (or value fn) at each moment of planning (even before the end of planning) are called ANYTIME Planning Alg.
- Planning allows to compute (contrast to 'learn') the best possible action.

(Model-based - Model-free)

- In a Model-Based setting, we can find out which (reward, next state) pairs are possible given current state and action.
- In a Model-Free setting, we know nothing about env dynamics. Optimization of agent decisions is based solely on sample-based experiences of the world.

(Types of Planning)

- Background Planning is synonymous to model-free learning from precollected samples of the given env model with the goal to improve policy-/value- fn in all the sampled states.

- Decision Time Planning starts after an agent's transition into a new state; it is used to select an optimal action for the current state only.

(Rollout Policy - Tree Policy)

- For long episodes and very short planning time RP should be as fast as possible, TP can be much slower.
- MCTS is relatively insensitive to the quality of RP. So we can use random RP, but it is always better to make RP as good as possible.

(MCTS)

- The Backup phase of MCTS performs Policy Evaluation; it makes value estimate in tree nodes consistent with compound policy (rollout + tree policy)
- The MAX action selection strategy (classic MCTS) is an instance of Policy Improvement; greedy action selection wrt action-value fn.
- MCTS aims to balance exploration and exploitation by treating action selection in each as an independent multi-armed bandit problem.